

BONUS

16 PAGE POCKET PROGRAMS




```

10020  UTAB 141 HTAB 141 PRINT "BY BLOCKS"
10030  GOSUB 20000
10040  UTAB 5
10050  PRINT "THERE ARE SIX BLOCKS, ALL OF DIFFERENT SIZES. THE PUZZLE ST
10060  ARTS OFF WITH THE BLOCKS DESIGNED IN A TOWER"
10070  PRINT "THE TOWER IS POSITIONED ON ONE OF THREE SITES. YOU M
10080  UST TRANSFER THE TOWER FROM ITS ORIGINAL SPOT TO ONE OF THE OTHER T
10090  W"
10100  PRINT "YOU MAY ONLY MOVE ONE BLOCK AT A TIME, AND YOU CANN
10110  OT MOVE ONE THAT IS LYING UNDER ANOTHER"
10120  PRINT "PRINT 'AT NO TIME CAN YOU MOVE A BLOCK ONTO A BLOCK OF SHM
10130  ALLER SIZE THAN ITSELF"
10140  GOSUB 20000
10150  UTAB 5
10160  PRINT "THE BLOCK YOU ARE CURRENTLY USING WILL BE ORANGE. TO CHANGE
10170  BLOCKS, YOU MAY PRESS EITHER 'H' OR 'L'"
10180  PRINT "H WILL SELECT A BLOCK THAT IS ONE SIZE SMALLER"
10190  PRINT "L WILL SELECT A BLOCK THAT IS ONE SIZE LARGER"
10200  PRINT "SWAP"
10210  PRINT "TO MOVE A BLOCK PRESS ONE OF THE ARROWS"
10220  PRINT "HIT SPACE BAR TO CONTINUE" IF PEEK C =
10230  130 THEN GOTO 10040
10240  POKE C = 10050: GOTO 10040
10250  POKE C = 10050: GOTO 10040

```

Machine Code For BASIC

By Derek Au

THIS PROGRAM decompiles a machine-code file into an Applesoft program consisting of POKEs and line numbers. This is useful for having machine-language sub-routines ready poked into memory from BASIC.

```

10  PRINT "MOMON C,I,O"
20  HOME
30  PRINT
40  INPUT "ENTER STARTING ADDRESS:?" AS:
   A = VAL (AS)
50  PRINT : PRINT
60  INPUT "ENTER ENDING ADDRESS:?" AS:
   X = VAL (XS)
80  DIM LINE(Z)
90  INPUT "YOUR DESIRED STARTING LINE:?" D
100 HOME
110 POKE 34,2: PRINT "SELECT MACHINE-CODE
   FILE"
120 PRINT "CATALOG"
130 INPUT "FILENAME:?" FILES
140 PRINT "BLOAD";FILES
150 POKE 34,0
160 HOME
170 PRINT "OPEN CODE"
180 PRINT "DELETE CODE"
190 PRINT "OPEN CODE"
200 PRINT "WRITE CODE"
210 PRINT "DEL0,400": FOR PR = A TO VAL (XS)
220 CO = CO + 1
230 IF CO = 10 THEN CO = 1
240 IF CO < > 1 THEN 200
250 PRINT
260 PRINT D: " "
270 D = D + 10
280 PRINT "POKE ";PR," "; PEEK (PR)," ";
290 NEXT PR
300 PRINT : PRINT "CLOSE CODE"
310 PRINT "MON C,I,O"
320 PRINT "EXEC CODE"
330 END

```

Ullo Again...

By S. Zanker

WITH THIS program, you will know how much random-access memory (RAM) your computer contains, and which slots have disk cards, without having to open the cover.

You will also know how much memory is available to an Applesoft BASIC program, and where that program will be located, and the number of buffers available for disk files (current value of MAXFILES).

The program variables contain this information:

DAY\$ = Date of creation (you supply)
 TIS\$ = Disk title (you supply)
 MEM = Highest RAM address + 1
 TY\$ = Type of disk (SLAVE/MASTER)
 TY\$ = Type of disk (SLAVE/MASTER)
 HI = Current value of HIMEM
 PR = Pointer to start of basic program

54

FR = Free memory for basic program
 VOL = Volume number of disk
 MF = Current value of MAXFILES
 SP\$ = Five spaces for editing

Line 20 gives you the option of recording the date the disk was initialised. Line 30 enables you to give the disk a title. There is a 40-character limit.

Line 50 disables the INIT command, preventing loss of non-write-protected disks.

Line 60 displays the amount of RAM measured in 1024 bytes.

Lines 70 and 80 display the type of disk last booted (SLAVE/MASTER), and the creation date.

Line 90 prints an inverse bar on the screen.

Lines 110 to 130 give a formatted display of the variables concerning BASIC program memory.

Lines 140 and 150 poke and execute the machine-language routine which looks for disk cards.

Another inverse line is printed at line 160. The screen is now divided into three sections: the top section deals with system data, the middle section with free memory and the lower section with disk information, which is displayed by lines 170 and 180.

Line 190 asks if you want a CATALOG. Your response will default to yes.

Line 200 issues the command, if you want a second greeting program, replace the statement with PRINT CHR\$(4) RUN filename'.

Line 210 rings a bell and wipes the program from memory, ready for use.

The program can be run on any size system. You'll be surprised how much RAM you have remaining after zero page, the input buffer, screen memory and DOS have all had their share.

```

10  REM GREETING PROGRAM
20  DAY$ = "01JUL82"
30  TITLES = "<> SUPPLY YOUR TITLE HERE <>"
40  MEM = ( PEEK (978) + 35) * 256:TY = ( PEEK
   (978) + 25) * 256 + 254:TT = (41 -
   LEN (TIS)) / 2:TT = TT + (TT < 1)
50  POKE MEM - 6833,96
60  TEXT : HOME : NORMAL : PRINT TAB (14);"APPLE
   |)+ "MEM / 1024;"K"
70  PRINT "DOS 3.3"; TAB (30);TY$ = "MASTER":
   IF PEEK (TY) < > 54 THEN TY$ = "SLAVE": FLASH
80  PRINT TY$;"DISK"; NORMAL : PRINT TAB(
   13);"CREATED ";DAY$
90  PRINT "INVERSE : PRINT TAB (40);" ": NORMAL
100 HI = PEEK (115) + PEEK (116) * 256:PR =
   PEEK (103) + PEEK (104) * 256:FR = HI - PR +
   1:VOL = PEEK (MEM - 2058):MF = PEEK (MEM -
   5545):SP$ = " "
110 PRINT "HIMEM=" ; RIGHTS (SP$ + STR$
   (HI),5)
120 PRINT "PROGRAM=" ; RIGHTS (SP$ + STR$
   (PR),5); TAB (28);"MAXFILES=" ;MF: PRINT TAB(
   10);"-----"
130 PRINT "FRE MEM=" ; RIGHTS (SP$ + STR$
   (FR),5);" (" ; INT (FR / 1024);"K)": PRINT
140 FOR A = 768 TO 810: READ B: POKE A,B: NEXT :
   DATA 162,0,169,200,134,6,133,7,160,7,198,7,
   165,7,201,192,240,24,177,6,217,1,251,208,239,
   136,136,16,245,165,7,233,192,32,227,253,32,
   72,249,76,8,3,96
150 PRINT "DISK CARDS ARE IN SLOT(S) " : CALL
   768: PRINT : PRINT
160 INVERSE : PRINT TAB (40);" ": NORMAL
170 PRINT TAB (13)"DISK VOLUME #";VOL: PRINT
180 PRINT TAB (TT);TIS: PRINT : PRINT
190 PRINT "CATALOG [Y] [N] ?[Y]": HTAB PEEK
   (36) - 1: POKE 49168,0: GET AN$: PRINT AN$
200 IF AN$ < > "N" THEN PRINT CHR$
   (4)"CATALOG"
210 CALL - 198: NEW

```

Epson Connections

By John Marquet

I HAVE an Apple II fitted out as a Pascal program development tool. The system has 48 kilobytes of memory, two disks, a 20-line by 80-character monitor and an Epson MS80F/T printer. Using this system, I have developed a program for printing one or more copies of a UCSD Pascal text file on an Epson printer, with an optional character density or size.

The UCSD Pascal text editor can handle and format simple correspondence as well as sourcing programs. When the dense-print option is selected on the printer, a double-s is given on each character, and the matrix print becomes clearer.

The program uses some of the features of the UCSD Pascal - for example, the file-naming conventions.

```

program typetext;
var hardcop,infile: text;
    numbercopy,indent,loop: integer;
    newk, innam: string;
    contr: char;

procedure initialise;
begin
    writeln;
    writeln('John's text output');
    writeln('Note that an input line with a leading " " will');
    writeln('result in a page throw on output. ');
    innam:= ' ';
    write('Type name of input text file: ');
    readln(innam);
    if pos(' ',innam)=0 then innam:=concat(innam,'.TEXT');
    reset(infile,innam);
    if ioresult<>0 then begin
        writeln('File not found');
        exit (program)
    end;
    contr:= ' ';
    writeln('Enter < for small, > for large print');
    writeln('else # for high density normal size print. ');
    writeln('Any other character gives normal size and density. ');
    write('Type of print? ');
    readln(contr);
    if contr = '<' then begin
        write('Offset (no. of char spaces in left margin) ');
        readln(indent);
    end else indent:=20;
    write('No. of copies? ');
    readln(numbercopy);
    if numbercopy<0 then exit (program);
    writeln('Set top of form on printer, then key "return"');
    readln(newk);
    rewrite(hardcop,'printer');

end;
begin
    while true do begin
        initialise;
        repeat begin
            reset(infile);
            while (not eof(infile)) do begin
                readln(infile,newk);
                writeln(newk);
                if pos(' ',newk)=1 then
                    begin
                        write(hardcop,chr(12));
                        newk[1]:= ' ';
                    end;
                begin
                    case contr of
                        '<': begin
                            write(hardcop,chr(15));
                            for loop:=1 to indent do write(hardcop,' ');
                        end;
                        '>': write(hardcop,chr(14));
                        '#': write (hardcop,chr(27),'E');
                    end;
                    writeln(hardcop,newk);
                end;
                case contr of
                    '<': write(hardcop,chr(18));
                    '>': write(hardcop,chr(27),'F');
                end;
            end;
            numbercopy:=numbercopy-1;
            if numbercopy>0 then write(hardcop,chr(12));
        end;
        until numbercopy=0;
        close(infile);
        close(hardcop);
    end;
end.

```

Hackatext

By Derek Au

HACKATEXT is a program which creates sequential files but allows you to input commas and colons which the MAKE TEXT program supplied on the System Master disk cannot - for example, BSAVE PROGRAM,A\$4000,L8192,S6,V100,D2.


```

10 REM MAKE TEXT
20 HOME
30 HTAB 10: PRINT "TEXT-FILE CREATOR#1"
40 HTAB 5: VTAB 7
50 PRINT " (M)AKE TEXT
   PRINT " (C)ATALOG
   PRINT " (E)XIT "
60 PRINT " WHICH?": GET B$
70 IF B$ = "M" THEN 110
80 IF B$ = "C" THEN PRINT: INPUT "PRESS
   'RETURN' TO CATALOG;NS: PRINT CHR$ (4)"CATALOG":
   INPUT "";NS: GOTO 20
90 IF B$ = "E" THEN HOME: PRINT "BYE!!": END
100 PRINT CHR$ (7): GOTO 20
110 PRINT: PRINT "MAKING TEXT: USE <CTRL>F TO
   ESCAPE.....<CTRL>R>RETURN..."
120 A = 8192
130 GET WS: PRINT WS;
140 IF WS = CHR$ (6) THEN 180
150 IF WS = CHR$ (18) THEN A = 8192: GOTO 20
160 POKE A, ASC (WS):A = A + 1
170 GOTO 130
180 INPUT "FINISHED?->":NS
190 IF LEFT$ (NS,1) = "Y" THEN 210
200 GOTO 130
210 INPUT "FILENAME?->":FILES
220 INPUT "(T)EXT-FILE OR (B)INARY-FILE?->":MS;
   IF MS = "B" THEN 320
230 IF MS = "T" THEN 250
240 PRINT CHR$ (7): VTAB PEEK (37): GOTO 220
250 PRINT CHR$ (4)"OPEN";FILES
260 PRINT CHR$ (4)"WRITE";FILES
270 FOR WR = 8192 TO A
280 PRINT CHR$ ( PEEK (WR));
290 NEXT
300 PRINT: PRINT CHR$ (4)"CLOSE";FILES
310 GOTO 20
320 PRINT CHR$ (4)"SAVE";FILES;"A8192,L";
   A = 8192
330 GOTO 20

```

Siren

By Robert Chalmers

HAVE YOU ever needed a short program that will sound an alarm for you at a pre-determined point in a main program?

Siren is a little machine-language program that will do just that. It is listed by code, hex and decimal, with a short demonstration program to show its use from within a larger framework. The starting address is 804 (dec), usually free for machine-code work.

```

10 FOR I = 804 TO 819
20 READ A
30 POKE I,A
40 NEXT I
50 DATA 160,0,169,106,153,208,7,32,58,255,200,192,40,200,243,96

```

To instigate this program from BASIC, use this method:

```

LDY #0      LOAD Y WITH 0
LDA #A      LOAD ACCUMULATOR WITH FLASHING *
STAS $700,Y STORE ACCUMULATOR AT $700 + CONTENTS OF Y.
              (BOTTOM LINE OF SCREEN.)
JSR $F3A    JUMP TO SUBROUTINE IN MONITOR THAT RINGS OR
              BELL
INY         INCREMENT Y REGISTER BY 1
CPY $28     COMPARE VALUE IN Y TO $28 (40 DEC)
BNE #3      BRANCH BACK 13 STEPS IF Y IS NOT EQUAL TO
              $28 (40)
RTS         RETURN TO MAIN PROGRAM

```

To demonstrate this in use, add the following:

```

60 PRINT "ENTER FIVE NUMBERS."
70 N=0:INPUT "INPUT A NUMBER.":B
80 LET N=N+1
90 PRINT B:IF N=5 THEN CALL 804
100 GOTO 70

```

Merry Christmas

By Derek Au

THIS PROGRAM plays the traditional "We Wish You A Merry Christmas" tune. Just some short programming fun! The program POKES some sound routines which can be used in your own programs. Just: POKE 768,(pitch 1-255) : POKE 769,(duration 1-255) : CALL 770.

```

10 GOSUB 110
20 HOME
30 AS = "WE WISH YOU A MERRY CHRISTMAS",
   WE WISH YOU A MERRY CHRISTMAS", WE WISH
   YOU A MERRY CHRISTMAS", ... AND A 'HAPPY
   NEW YEAR!' "
40 DATA 133,110, 98,99, 98,99,88,80, 98,100,105,110,
   117,130, 117,150, 117,150, 88,110,
   88,110, 78,75, 88,99, 98,99, 105,150, 105,150,
   105,160, 78,100, 78,100, 67,110
50 DATA 78,100,88,100,98,100, 98,100, 133,130,
   133,130, 117,115, 88,120, 105,120, 98,200
60 FOR R = 1 TO 30
70 READ P,D
80 POKE 768,P: POKE 769,D: CALL 770
90 NEXT
100 END
110 POKE 768,0: POKE 769,173: POKE 770,173: POKE
   771,48: POKE 772,192: POKE 773,136: POKE 774,208:
   POKE 775,5: POKE 776,206:
120 POKE 777,1: POKE 778,3: POKE 779,240: POKE
   780,9: POKE 781,202: POKE 782,208: POKE 783,245:
   POKE 784,174: POKE 785,0:
130 POKE 786,3: POKE 787,76: POKE 788,2: POKE
   789,3: POKE 790,96: POKE 791,0:
140 RETURN

```

OUR COURIERS DELIVER TO YOUR DOOR

ORDER HOTLINE (02) 232-7704

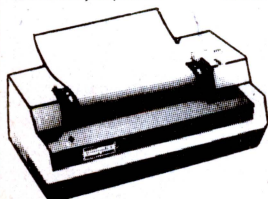
- ★ **BANKCARD**
- ★ **MOST ORDERS SHIPPED WITHIN 48 HRS**
- ★ **14-DAY MONEY BACK GUARANTEE**
- ★ **WE WILL TRY TO BEAT ANY CURRENTLY ADVERTISED PRICE**
- ★ **WRITE TO: G.P.O. BOX 4475, SYDNEY, N.S.W. 2001.**

COMPUTER DISCOUNTERS



MAIL ORDER ADVERTISING PRICE LIST

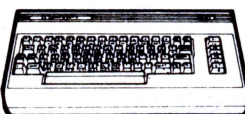
Vic 20 - Priced too low to list
 Vic 1541 Single Disk Drive 599.00
 Vic Datassette 84.00
 Joystick Controller 16.00
 Vic 1515 Graphics Printer 399.00
 Vic Super Expander 59.00
 3K Memory Expansion 45.00
 8K Memory Expansion 69.00
 16K Memory Expansion 110.00



Software for VIC 20
 Games Pack 1 (3 programs) 16.95
 Games Pack 2 (3 programs) 19.95
 Ludwig's Lemon Laser 16.95
 Kongo Kong 19.95
 Annihilator 19.95
 Trek 16.95
 Bug Blast 19.95
 Headon 17.95
 Paratrooper 19.95
 Cricket 16.95
 Artillery 16.95
 Alien Invasion 19.95
 Adventurer Pack (8k or 16k) 19.95
 Allied Defense 19.95
 Mail it 20 19.95
 Micro Hex Assembler Editor (3k or 8k) 19.95
 Cosmic Crystals (paddles) 19.95
 Rebel Defender (8k & paddles) 19.95
 Cosmic Crusader 19.95
 Blasteroids 16.95
 Ultimate Tank 3k 19.95
 Champ Chase 16.95
 Cosmic Crossfire 19.95
 Cheque Book (3k, 8k or 16k) 24.95
 Horse File (3k, 8k or 16k) 24.95
 Grave Robbers 19.95
 Dungeon Quest (16k) 19.95
 Adventure Pack 1 (3 programs) 19.95
 Adventure Pack 2 (3 programs) 19.95
 4 in one Games Special 19.95

All Commodore VIC 20 games cartridges 35.00
 Creative Software cartridge for VIC 20
 Choplifter, Trashman, Serpentine, Astra Blitz, Apple Panic, All 49.95
 3 slot memory expansion board now with on/off and reset switch 59.95
 Light pens now available for VIC 20
 CALL FOR PRICE

COMMODORE 64 CALL FOR PRICE



COMMODORE 64 SOFTWARE
 Grave Robbers 19.95
 Annihilator 24.95
 Trek 16.95
 Adventurer Pack 1 (3 programs) 19.95
 Adventurer Pack 2 (3 programs) 19.95

HARDWARE
 Atari 800 Computer 48K with Basic 1225.00
 Atari 400 Computer 48K 639.00
 Atari 410 Program Recorder 139.00
 Atari 810 Disk Drive 769.00
 Atari Joystick per Pair 36.00

SOFTWARE
 Wordprocessor (Disk) 289.00
 Personal Financial Management (D) 149.00
 Conversational French (T) 79.00
 Conversational German (T) 79.00
 Conversational Spanish (T) 79.00
 Conversational Italian (T) 79.00
 Pacman (Cart) 65.00
 Centipede (C) 65.00
 Star Raiders (C) 65.00
 Space Invaders (C) 53.00
 Missile Command (C) 53.00
 Asteroids (C) 53.00
 Apple Panic (C or D) 39.95



KAYPRO

KAYPRO II COMPUTER Including:
 SBasic C.P.M. Profit Plan, Perfect Calc, Perfect Writer, Perfect Speller and Perfect Filer
 64K Memory 8" Monitor 200K per disk fully portable PHONE FOR OUR VERY SPECIAL PRICE!!



PRINTERS

Brother HRI DaisyWheel 15" 1299.00
 CITHO 8510 80 Col Dot Matrix 995.00
 CITHO1550 132 Col DotMatrix 1195.00
 CITHO F1040 Daisy Wheel 15" 2095.00
 Star Printer Centronics Parallel 529.00

FREIGHT CHARGES

Under 5KG - ADD \$5.00
 5-10KG - ADD \$7.50
 PRICES SUBJECT TO CHANGE

TANDY TRS80, SYSTEM 80

Artist

By Tony Hinde

ARTIST is a program that stemmed from a simple desire to please my five-year-old neighbour. To draw anything on the screen, you use the four arrow keys. To get a diagonal line, you must press the appropriate two keys – for example, up and left will draw a line toward the top left-hand corner of the screen. CLEAR will clear the screen.

If you don't own a GP-80, simply delete line 135 and from the remark line onward. Once you're satisfied with your drawing, press the space bar and it will print a 4.4 cm by 3.8 mm copy on to the printer.

You can add functions to the program, like one to save the screen data to tape for later use.

```
10 CLS:
20 CLEAR200
30 DEFINITR=Z
35 X=63:Y=23
40 PEEK(15359) SET(X,Y):RESET(X,Y):IF#0THEN40
45 SET(X,Y)
50 IF#0THENY=Y-1
60 IF#15THENY=Y+1
70 IF#32THENX=X-1
80 IF#64THENX=X+1
90 IF#2THENCLS
100 IF#72THENY=Y-1:X=X+1
110 IF#48THENY=Y+1:X=X-1
120 IF#88THENX=X+1:Y=Y-1
130 IF#48THENY=Y-1:X=X-1
135 IF#128THEN200
140 IF#127THENX=X+127
150 IF#0THENX=0
160 IF#127THENY=47
170 IF#0THENY=0
180 GOTD40
200 * This gives a hard copy of the screen on a GP-80 Printer.
210 CLEAR1000
220 DIMA(128)
225 LPRINTCHR$(8)
230 FORX=0TO127
235 Y=0
237 IFY=47THEN40
240 R=POINT(X,Y):X=X-1
250 M=M+R*3
260 Y=Y+1:R=POINT(X,Y):X=X-1:M=M+R*12
270 Y=Y+1:R=POINT(X,Y):X=X-1:M=M+R*48
280 Y=Y+1:R=POINT(X,Y):X=X-1:M=M+R*64:K=K+R
290 Y=Y+1:R=POINT(X,Y):X=X-1:K=K+R*6
300 Y=Y+1:R=POINT(X,Y):X=X-1:K=K+R*24
310 Y=Y+1:IFY=48THENR=POINT(X,Y):X=X-1:K=K+R*96
320 M=M+128:K=K+128
330 RX=X
340 LPRINTCHR$(M)
350 M=0:K=0
360 NEXTX
370 LPRINT
380 FORX=0TO127:LPRINTCHR$(R(X)):NEXT
390 LPRINT
400 P=P+7
410 GOTD200
```

(Model 3 with printer):

Check Your Cheques

By R.C. Heslewood

THIS PROGRAM will store and access cheque records as a complete listing or, if hard copy is required, will output them to a line printer. Though it was written for cheque records, it can be used for cash or other types of transactions.

The program will run on a machine with 16 kilobytes of RAM, and handle up to 250 entries and 20 categories. Those who have larger memory capacities can increase the values of the variables on line 30: N = number of entries and C = number of categories. The

CLEAR statement on line 10 will need increasing for more string space.

Suggested changes are:

32 kilobytes RAM for 650 entries and 30 categories:

Line 10 CLS: CLEAR 14500

Line 30 C = 30: N = 650: S = 200: SC = 50

48 kilobytes RAM for 1100 entries and 50 categories:

Line 10 CLS: CLEAR 24500

Line 30 C = 50: N = 1100: S = 200: SC = 50

The other variables on line 30 may also be of interest: S = Initial scrolling speed; SC = Amount of change to scrolling speed with each press of the S or F keys.

The S and F keys are used to decrease and increase the scrolling speed while the information is being listed. The display can also be temporarily stopped and re-started again.

Of interest to anyone wishing to convert to another type of computer is the POKE 16916,3 statement – the scroll protects the top three lines during listing. POKE 16916,0 cancels the scroll protect. These statements will probably have their equivalent for other machines.

The program will, in its present form, run on machines with or without line printers. However, anyone not having a printer could delete the printer references without making major changes to the program.

```
1 * CHEQUE RECORDS
2
3 CLS: CLEAR2000
4 DEFINITR=Z
5 C=20:N=250:S=200:SC=50
6 DIM J%(C),P%(N),C%(N),D%(N),M%(N),T%(N)
7 M=PAYEE
8 M=PAYEE
9 M=PAYEE
10 M=PAYEE
11 M=PAYEE
12 M=PAYEE
13 M=PAYEE
14 M=PAYEE
15 M=PAYEE
16 M=PAYEE
17 M=PAYEE
18 M=PAYEE
19 M=PAYEE
20 M=PAYEE
21 M=PAYEE
22 M=PAYEE
23 M=PAYEE
24 M=PAYEE
25 M=PAYEE
26 M=PAYEE
27 M=PAYEE
28 M=PAYEE
29 M=PAYEE
30 M=PAYEE
31 M=PAYEE
32 M=PAYEE
33 M=PAYEE
34 M=PAYEE
35 M=PAYEE
36 M=PAYEE
37 M=PAYEE
38 M=PAYEE
39 M=PAYEE
40 M=PAYEE
41 M=PAYEE
42 M=PAYEE
43 M=PAYEE
44 M=PAYEE
45 M=PAYEE
46 M=PAYEE
47 M=PAYEE
48 M=PAYEE
49 M=PAYEE
50 M=PAYEE
51 M=PAYEE
52 M=PAYEE
53 M=PAYEE
54 M=PAYEE
55 M=PAYEE
56 M=PAYEE
57 M=PAYEE
58 M=PAYEE
59 M=PAYEE
60 M=PAYEE
61 M=PAYEE
62 M=PAYEE
63 M=PAYEE
64 M=PAYEE
65 M=PAYEE
66 M=PAYEE
67 M=PAYEE
68 M=PAYEE
69 M=PAYEE
70 M=PAYEE
71 M=PAYEE
72 M=PAYEE
73 M=PAYEE
74 M=PAYEE
75 M=PAYEE
76 M=PAYEE
77 M=PAYEE
78 M=PAYEE
79 M=PAYEE
80 M=PAYEE
81 M=PAYEE
82 M=PAYEE
83 M=PAYEE
84 M=PAYEE
85 M=PAYEE
86 M=PAYEE
87 M=PAYEE
88 M=PAYEE
89 M=PAYEE
90 M=PAYEE
91 M=PAYEE
92 M=PAYEE
93 M=PAYEE
94 M=PAYEE
95 M=PAYEE
96 M=PAYEE
97 M=PAYEE
98 M=PAYEE
99 M=PAYEE
100 M=PAYEE
101 M=PAYEE
102 M=PAYEE
103 M=PAYEE
104 M=PAYEE
105 M=PAYEE
106 M=PAYEE
107 M=PAYEE
108 M=PAYEE
109 M=PAYEE
110 M=PAYEE
111 M=PAYEE
112 M=PAYEE
113 M=PAYEE
114 M=PAYEE
115 M=PAYEE
116 M=PAYEE
117 M=PAYEE
118 M=PAYEE
119 M=PAYEE
120 M=PAYEE
121 M=PAYEE
122 M=PAYEE
123 M=PAYEE
124 M=PAYEE
125 M=PAYEE
126 M=PAYEE
127 M=PAYEE
128 M=PAYEE
129 M=PAYEE
130 M=PAYEE
131 M=PAYEE
132 M=PAYEE
133 M=PAYEE
134 M=PAYEE
135 M=PAYEE
136 M=PAYEE
137 M=PAYEE
138 M=PAYEE
139 M=PAYEE
140 M=PAYEE
141 M=PAYEE
142 M=PAYEE
143 M=PAYEE
144 M=PAYEE
145 M=PAYEE
146 M=PAYEE
147 M=PAYEE
148 M=PAYEE
149 M=PAYEE
150 M=PAYEE
151 M=PAYEE
152 M=PAYEE
153 M=PAYEE
154 M=PAYEE
155 M=PAYEE
156 M=PAYEE
157 M=PAYEE
158 M=PAYEE
159 M=PAYEE
160 M=PAYEE
161 M=PAYEE
162 M=PAYEE
163 M=PAYEE
164 M=PAYEE
165 M=PAYEE
166 M=PAYEE
167 M=PAYEE
168 M=PAYEE
169 M=PAYEE
170 M=PAYEE
171 M=PAYEE
172 M=PAYEE
173 M=PAYEE
174 M=PAYEE
175 M=PAYEE
176 M=PAYEE
177 M=PAYEE
178 M=PAYEE
179 M=PAYEE
180 M=PAYEE
181 M=PAYEE
182 M=PAYEE
183 M=PAYEE
184 M=PAYEE
185 M=PAYEE
186 M=PAYEE
187 M=PAYEE
188 M=PAYEE
189 M=PAYEE
190 M=PAYEE
191 M=PAYEE
192 M=PAYEE
193 M=PAYEE
194 M=PAYEE
195 M=PAYEE
196 M=PAYEE
197 M=PAYEE
198 M=PAYEE
199 M=PAYEE
200 M=PAYEE
201 M=PAYEE
202 M=PAYEE
203 M=PAYEE
204 M=PAYEE
205 M=PAYEE
206 M=PAYEE
207 M=PAYEE
208 M=PAYEE
209 M=PAYEE
210 M=PAYEE
211 M=PAYEE
212 M=PAYEE
213 M=PAYEE
214 M=PAYEE
215 M=PAYEE
216 M=PAYEE
217 M=PAYEE
218 M=PAYEE
219 M=PAYEE
220 M=PAYEE
221 M=PAYEE
222 M=PAYEE
223 M=PAYEE
224 M=PAYEE
225 M=PAYEE
226 M=PAYEE
227 M=PAYEE
228 M=PAYEE
229 M=PAYEE
230 M=PAYEE
231 M=PAYEE
232 M=PAYEE
233 M=PAYEE
234 M=PAYEE
235 M=PAYEE
236 M=PAYEE
237 M=PAYEE
238 M=PAYEE
239 M=PAYEE
240 M=PAYEE
241 M=PAYEE
242 M=PAYEE
243 M=PAYEE
244 M=PAYEE
245 M=PAYEE
246 M=PAYEE
247 M=PAYEE
248 M=PAYEE
249 M=PAYEE
250 M=PAYEE
251 M=PAYEE
252 M=PAYEE
253 M=PAYEE
254 M=PAYEE
255 M=PAYEE
256 M=PAYEE
257 M=PAYEE
258 M=PAYEE
259 M=PAYEE
260 M=PAYEE
261 M=PAYEE
262 M=PAYEE
263 M=PAYEE
264 M=PAYEE
265 M=PAYEE
266 M=PAYEE
267 M=PAYEE
268 M=PAYEE
269 M=PAYEE
270 M=PAYEE
271 M=PAYEE
272 M=PAYEE
273 M=PAYEE
274 M=PAYEE
275 M=PAYEE
276 M=PAYEE
277 M=PAYEE
278 M=PAYEE
279 M=PAYEE
280 M=PAYEE
281 M=PAYEE
282 M=PAYEE
283 M=PAYEE
284 M=PAYEE
285 M=PAYEE
286 M=PAYEE
287 M=PAYEE
288 M=PAYEE
289 M=PAYEE
290 M=PAYEE
291 M=PAYEE
292 M=PAYEE
293 M=PAYEE
294 M=PAYEE
295 M=PAYEE
296 M=PAYEE
297 M=PAYEE
298 M=PAYEE
299 M=PAYEE
300 M=PAYEE
301 M=PAYEE
302 M=PAYEE
303 M=PAYEE
304 M=PAYEE
305 M=PAYEE
306 M=PAYEE
307 M=PAYEE
308 M=PAYEE
309 M=PAYEE
310 M=PAYEE
311 M=PAYEE
312 M=PAYEE
313 M=PAYEE
314 M=PAYEE
315 M=PAYEE
316 M=PAYEE
317 M=PAYEE
318 M=PAYEE
319 M=PAYEE
320 M=PAYEE
321 M=PAYEE
322 M=PAYEE
323 M=PAYEE
324 M=PAYEE
325 M=PAYEE
326 M=PAYEE
327 M=PAYEE
328 M=PAYEE
329 M=PAYEE
330 M=PAYEE
331 M=PAYEE
332 M=PAYEE
333 M=PAYEE
334 M=PAYEE
335 M=PAYEE
336 M=PAYEE
337 M=PAYEE
338 M=PAYEE
339 M=PAYEE
340 M=PAYEE
341 M=PAYEE
342 M=PAYEE
343 M=PAYEE
344 M=PAYEE
345 M=PAYEE
346 M=PAYEE
347 M=PAYEE
348 M=PAYEE
349 M=PAYEE
350 M=PAYEE
351 M=PAYEE
352 M=PAYEE
353 M=PAYEE
354 M=PAYEE
355 M=PAYEE
356 M=PAYEE
357 M=PAYEE
358 M=PAYEE
359 M=PAYEE
360 M=PAYEE
361 M=PAYEE
362 M=PAYEE
363 M=PAYEE
364 M=PAYEE
365 M=PAYEE
366 M=PAYEE
367 M=PAYEE
368 M=PAYEE
369 M=PAYEE
370 M=PAYEE
371 M=PAYEE
372 M=PAYEE
373 M=PAYEE
374 M=PAYEE
375 M=PAYEE
376 M=PAYEE
377 M=PAYEE
378 M=PAYEE
379 M=PAYEE
380 M=PAYEE
381 M=PAYEE
382 M=PAYEE
383 M=PAYEE
384 M=PAYEE
385 M=PAYEE
386 M=PAYEE
387 M=PAYEE
388 M=PAYEE
389 M=PAYEE
390 M=PAYEE
391 M=PAYEE
392 M=PAYEE
393 M=PAYEE
394 M=PAYEE
395 M=PAYEE
396 M=PAYEE
397 M=PAYEE
398 M=PAYEE
399 M=PAYEE
400 M=PAYEE
401 M=PAYEE
402 M=PAYEE
403 M=PAYEE
404 M=PAYEE
405 M=PAYEE
406 M=PAYEE
407 M=PAYEE
408 M=PAYEE
409 M=PAYEE
410 M=PAYEE
411 M=PAYEE
412 M=PAYEE
413 M=PAYEE
414 M=PAYEE
415 M=PAYEE
416 M=PAYEE
417 M=PAYEE
418 M=PAYEE
419 M=PAYEE
420 M=PAYEE
421 M=PAYEE
422 M=PAYEE
423 M=PAYEE
424 M=PAYEE
425 M=PAYEE
426 M=PAYEE
427 M=PAYEE
428 M=PAYEE
429 M=PAYEE
430 M=PAYEE
431 M=PAYEE
432 M=PAYEE
433 M=PAYEE
434 M=PAYEE
435 M=PAYEE
436 M=PAYEE
437 M=PAYEE
438 M=PAYEE
439 M=PAYEE
440 M=PAYEE
441 M=PAYEE
442 M=PAYEE
443 M=PAYEE
444 M=PAYEE
445 M=PAYEE
446 M=PAYEE
447 M=PAYEE
448 M=PAYEE
449 M=PAYEE
450 M=PAYEE
451 M=PAYEE
452 M=PAYEE
453 M=PAYEE
454 M=PAYEE
455 M=PAYEE
456 M=PAYEE
457 M=PAYEE
458 M=PAYEE
459 M=PAYEE
460 M=PAYEE
461 M=PAYEE
462 M=PAYEE
463 M=PAYEE
464 M=PAYEE
465 M=PAYEE
466 M=PAYEE
467 M=PAYEE
468 M=PAYEE
469 M=PAYEE
470 M=PAYEE
471 M=PAYEE
472 M=PAYEE
473 M=PAYEE
474 M=PAYEE
475 M=PAYEE
476 M=PAYEE
477 M=PAYEE
478 M=PAYEE
479 M=PAYEE
480 M=PAYEE
481 M=PAYEE
482 M=PAYEE
483 M=PAYEE
484 M=PAYEE
485 M=PAYEE
486 M=PAYEE
487 M=PAYEE
488 M=PAYEE
489 M=PAYEE
490 M=PAYEE
491 M=PAYEE
492 M=PAYEE
493 M=PAYEE
494 M=PAYEE
495 M=PAYEE
496 M=PAYEE
497 M=PAYEE
498 M=PAYEE
499 M=PAYEE
500 M=PAYEE
501 M=PAYEE
502 M=PAYEE
503 M=PAYEE
504 M=PAYEE
505 M=PAYEE
506 M=PAYEE
507 M=PAYEE
508 M=PAYEE
509 M=PAYEE
510 M=PAYEE
511 M=PAYEE
512 M=PAYEE
513 M=PAYEE
514 M=PAYEE
515 M=PAYEE
516 M=PAYEE
517 M=PAYEE
518 M=PAYEE
519 M=PAYEE
520 M=PAYEE
521 M=PAYEE
522 M=PAYEE
523 M=PAYEE
524 M=PAYEE
525 M=PAYEE
526 M=PAYEE
527 M=PAYEE
528 M=PAYEE
529 M=PAYEE
530 M=PAYEE
531 M=PAYEE
532 M=PAYEE
533 M=PAYEE
534 M=PAYEE
535 M=PAYEE
536 M=PAYEE
537 M=PAYEE
538 M=PAYEE
539 M=PAYEE
540 M=PAYEE
541 M=PAYEE
542 M=PAYEE
543 M=PAYEE
544 M=PAYEE
545 M=PAYEE
546 M=PAYEE
547 M=PAYEE
548 M=PAYEE
549 M=PAYEE
550 M=PAYEE
551 M=PAYEE
552 M=PAYEE
553 M=PAYEE
554 M=PAYEE
555 M=PAYEE
556 M=PAYEE
557 M=PAYEE
558 M=PAYEE
559 M=PAYEE
560 M=PAYEE
561 M=PAYEE
562 M=PAYEE
563 M=PAYEE
564 M=PAYEE
565 M=PAYEE
566 M=PAYEE
567 M=PAYEE
568 M=PAYEE
569 M=PAYEE
570 M=PAYEE
571 M=PAYEE
572 M=PAYEE
573 M=PAYEE
574 M=PAYEE
575 M=PAYEE
576 M=PAYEE
577 M=PAYEE
578 M=PAYEE
579 M=PAYEE
580 M=PAYEE
581 M=PAYEE
582 M=PAYEE
583 M=PAYEE
584 M=PAYEE
585 M=PAYEE
586 M=PAYEE
587 M=PAYEE
588 M=PAYEE
589 M=PAYEE
590 M=PAYEE
591 M=PAYEE
592 M=PAYEE
593 M=PAYEE
594 M=PAYEE
595 M=PAYEE
596 M=PAYEE
597 M=PAYEE
598 M=PAYEE
599 M=PAYEE
600 M=PAYEE
601 M=PAYEE
602 M=PAYEE
603 M=PAYEE
604 M=PAYEE
605 M=PAYEE
606 M=PAYEE
607 M=PAYEE
608 M=PAYEE
609 M=PAYEE
610 M=PAYEE
611 M=PAYEE
612 M=PAYEE
613 M=PAYEE
614 M=PAYEE
615 M=PAYEE
616 M=PAYEE
617 M=PAYEE
618 M=PAYEE
619 M=PAYEE
620 M=PAYEE
621 M=PAYEE
622 M=PAYEE
623 M=PAYEE
624 M=PAYEE
625 M=PAYEE
626 M=PAYEE
627 M=PAYEE
628 M=PAYEE
629 M=PAYEE
630 M=PAYEE
631 M=PAYEE
632 M=PAYEE
633 M=PAYEE
634 M=PAYEE
635 M=PAYEE
636 M=PAYEE
637 M=PAYEE
638 M=PAYEE
639 M=PAYEE
640 M=PAYEE
641 M=PAYEE
642 M=PAYEE
643 M=PAYEE
644 M=PAYEE
645 M=PAYEE
646 M=PAYEE
647 M=PAYEE
648 M=PAYEE
649 M=PAYEE
650 M=PAYEE
651 M=PAYEE
652 M=PAYEE
653 M=PAYEE
654 M=PAYEE
655 M=PAYEE
656 M=PAYEE
657 M=PAYEE
658 M=PAYEE
659 M=PAYEE
660 M=PAYEE
661 M=PAYEE
662 M=PAYEE
663 M=PAYEE
664 M=PAYEE
665 M=PAYEE
666 M=PAYEE
667 M=PAYEE
668 M=PAYEE
669 M=PAYEE
670 M=PAYEE
671 M=PAYEE
672 M=PAYEE
673 M=PAYEE
674 M=PAYEE
675 M=PAYEE
676 M=PAYEE
677 M=PAYEE
678 M=PAYEE
679 M=PAYEE
680 M=PAYEE
681 M=PAYEE
682 M=PAYEE
683 M=PAYEE
684 M=PAYEE
685 M=PAYEE
686 M=PAYEE
687 M=PAYEE
688 M=PAYEE
689 M=PAYEE
690 M=PAYEE
691 M=PAYEE
692 M=PAYEE
693 M=PAYEE
694 M=PAYEE
695 M=PAYEE
696 M=PAYEE
697 M=PAYEE
698 M=PAYEE
699 M=PAYEE
700 M=PAYEE
701 M=PAYEE
702 M=PAYEE
703 M=PAYEE
704 M=PAYEE
705 M=PAYEE
706 M=PAYEE
707 M=PAYEE
708 M=PAYEE
709 M=PAYEE
710 M=PAYEE
711 M=PAYEE
712 M=PAYEE
713 M=PAYEE
714 M=PAYEE
715 M=PAYEE
716 M=PAYEE
717 M=PAYEE
718 M=PAYEE
719 M=PAYEE
720 M=PAYEE
721 M=PAYEE
722 M=PAYEE
723 M=PAYEE
724 M=PAYEE
725 M=PAYEE
726 M=PAYEE
727 M=PAYEE
728 M=PAYEE
729 M=PAYEE
730 M=PAYEE
731 M=PAYEE
732 M=PAYEE
733 M=PAYEE
734 M=PAYEE
735 M=PAYEE
736 M=PAYEE
737 M=PAYEE
738 M=PAYEE
739 M=PAYEE
740 M=PAYEE
741 M=PAYEE
742 M=PAYEE
743 M=PAYEE
744 M=PAYEE
745 M=PAYEE
746 M=PAYEE
747 M=PAYEE
748 M=PAYEE
749 M=PAYEE
750 M=PAYEE
751 M=PAYEE
752 M=PAYEE
753 M=PAYEE
754 M=PAYEE
755 M=PAYEE
756 M=PAYEE
757 M=PAYEE
758 M=PAYEE
759 M=PAYEE
760 M=PAYEE
761 M=PAYEE
762 M=PAYEE
763 M=PAYEE
764 M=PAYEE
765 M=PAYEE
766 M=PAYEE
767 M=PAYEE
768 M=PAYEE
769 M=PAYEE
770 M=PAYEE
771 M=PAYEE
772 M=PAYEE
773 M=PAYEE
774 M=PAYEE
775 M=PAYEE
776 M=PAYEE
777 M=PAYEE
778 M=PAYEE
779 M=PAYEE
780 M=PAYEE
781 M=PAYEE
782 M=PAYEE
783 M=PAYEE
784 M=PAYEE
785 M=PAYEE
786 M=PAYEE
787 M=PAYEE
788 M=PAYEE
789 M=PAYEE
790 M=PAYEE
791 M=PAYEE
792 M=PAYEE
793 M=PAYEE
794 M=PAYEE
795 M=PAYEE
796 M=PAYEE
797 M=PAYEE
798 M=PAYEE
799 M=PAYEE
800 M=PAYEE
801 M=PAYEE
802 M=PAYEE
803 M=PAYEE
804 M=PAYEE
805 M=PAYEE
806 M=PAYEE
807 M=PAYEE
808 M=PAYEE
809 M=PAYEE
810 M=PAYEE
811 M=PAYEE
812 M=PAYEE
813 M=PAYEE
814 M=PAYEE
815 M=PAYEE
816 M=PAYEE
817 M=PAYEE
818 M=PAYEE
819 M=PAYEE
820 M=PAYEE
821 M=PAYEE
822 M=PAYEE
823 M=PAYEE
824 M=PAYEE
825 M=PAYEE
826 M=PAYEE
827 M=PAYEE
828 M=PAYEE
829 M=PAYEE
830 M=PAYEE
831 M=PAYEE
832 M=PAYEE
833 M=PAYEE
834 M=PAYEE
835 M=PAYEE
836 M=PAYEE
837 M=PAYEE
838 M=PAYEE
839 M=PAYEE
840 M=PAYEE
841 M=PAYEE
842 M=PAYEE
843 M=PAYEE
844 M=PAYEE
845 M=PAYEE
846 M=PAYEE
847 M=PAYEE
848 M=PAYEE
849 M=PAYEE
850 M=PAYEE
851 M=PAYEE
852 M=PAYEE
853 M=PAYEE
854 M=PAYEE
855 M=PAYEE
856 M=PAYEE
857 M=PAYEE
858 M=PAYEE
859 M=PAYEE
860 M=PAYEE
861 M=PAYEE
862 M=PAYEE
863 M=PAYEE
864 M=PAYEE
865 M=PAYEE
866 M=PAYEE
867 M=PAYEE
868 M=PAYEE
869 M=PAYEE
870 M=PAYEE
871 M=PAYEE
872 M=PAYEE
873 M=PAYEE
874 M=PAYEE
875 M=PAYEE
876 M=PAYEE
877 M=PAYEE
878 M=PAYEE
879 M=PAYEE
880 M=PAYEE
881 M=PAYEE
882 M=PAYEE
883 M=PAYEE
884 M=PAYEE
885 M=PAYEE
886 M=PAYEE
887 M=PAYEE
888 M=PAYEE
889 M=PAYEE
890 M=PAYEE
891 M=PAYEE
892 M=PAYEE
893 M=PAYEE
894 M=PAYEE
895 M=PAYEE
896 M=PAYEE
897 M=PAYEE
898 M=PAYEE
899 M=PAYEE
900 M=PAYEE
901 M=PAYEE
902 M=PAYEE
903 M=PAYEE
904 M=PAYEE
905 M=PAYEE
906 M=PAYEE
907 M=PAYEE
908 M=PAYEE
909 M=PAYEE
910 M=PAYEE
911 M=PAYEE
912 M=PAYEE
913 M=PAYEE
914 M=PAYEE
915 M=PAYEE
916 M=PAYEE
917 M=PAYEE
918 M=PAYEE
919 M=PAYEE
920 M=PAYEE
921 M=PAYEE
922 M=PAYEE
923 M=PAYEE
924 M=PAYEE
925 M=PAYEE
926 M=PAYEE
927 M=PAYEE
928 M=PAYEE
929 M=PAYEE
930 M=PAYEE
931 M=PAYEE
932 M=PAYEE
933 M=PAYEE
934 M=PAYEE
935 M=PAYEE
936 M=PAYEE
937 M=PAYEE
938 M=PAYEE
939 M=PAYEE
940 M=PAYEE
941 M=PAYEE
942 M=PAYEE
943 M=PAYEE
944 M=PAYEE
945 M=PAYEE
946 M=PAYEE
947 M=PAYEE
948 M=PAYEE
949 M=PAYEE
950 M=PAYEE
951 M=PAYEE
952 M=PAYEE
953 M=PAYEE
954 M=PAYEE
955 M=PAYEE
956 M=PAYEE
957 M=PAYEE
958 M=PAYEE
959 M=PAYEE
960 M=PAYEE
961 M=PAYEE
962 M=PAYEE
963 M=PAYEE
964 M=PAYEE
965 M=PAYEE
966 M=PAYEE
967 M=PAYEE
968 M=PAYEE
969 M=PAYEE
970 M=PAYEE
971 M=PAYEE
972 M=PAYEE
973 M=PAYEE
974 M=PAYEE
975 M=PAYEE
976 M=PAYEE
977 M=PAYEE
978 M=PAYEE
979 M=PAYEE
980 M=PAYEE
981 M=PAYEE
982 M=PAYEE
983 M=PAYEE
984 M=PAYEE
985 M=PAYEE
986 M=PAYEE
987 M=PAYEE
988 M=PAYEE
989 M=PAYEE
990 M=PAYEE
991 M=PAYEE
992 M=PAYEE
993 M=PAYEE
994 M=PAYEE
995 M=PAYEE
996 M=PAYEE
997 M=PAYEE
998 M=PAYEE
999 M=PAYEE
1000 M=PAYEE
1001 M=PAYEE
1002 M=PAYEE
1003 M=PAYEE
1004 M=PAYEE
1005 M=PAYEE
1006 M=PAYEE
1007 M=PAYEE
1008 M=PAYEE
1009 M=PAYEE
1010 M=PAYEE
1011 M=PAYEE
1012 M=PAYEE
1013 M=PAYEE
1014 M=PAYEE
1015 M=PAYEE
1016 M=PAYEE
1017 M=PAYEE
1018 M=PAYEE
1019 M=PAYEE
1020 M=PAYEE
1021 M=PAYEE
1022 M=PAYEE
1023 M=PAYEE
1024 M=PAYEE
1025 M=PAYEE
1026 M=PAYEE
1027 M=PAYEE
1028 M=PAYEE
1029 M=PAYEE
1030 M=PAYEE
1031 M=PAYEE
1032 M=PAYEE
1033 M=PAYEE
1034 M=PAYEE
1035 M=PAYEE
1036 M=PAYEE
1037 M=PAYEE
1038 M=PAYEE
1039 M=PAYEE
1040 M=PAYEE
1041 M=PAYEE
1042 M=PAYEE
1043 M=PAYEE
1044 M=PAYEE
1045 M=PAYEE
1046 M=PAYEE
1047 M=PAYEE
1048 M=PAYEE
1049 M=PAYEE
1050 M=PAYEE
1051 M=PAYEE
1052 M=PAYEE
1053 M=PAYEE
1054 M=PAYEE
1055 M=PAYEE
1056 M=PAYEE
1057 M=PAYEE
1058 M=PAYEE
1059 M=PAYEE
1060 M=PAYEE
1061 M=PAYEE
1062 M=PAYEE
1063 M=PAYEE
1064 M=PAYEE
1065 M=PAYEE
1066 M=PAYEE
1067 M=PAYEE
1068 M=PAYEE
1069 M=PAYEE
1070 M=PAYEE
1071 M=PAYEE
1072 M=PAYEE
1073 M=PAYEE
1074 M=PAYEE
1075 M=PAYEE
1076 M=PAYEE
1077 M=PAYEE
1078 M=PAYEE
1079 M=PAYEE
1080 M=PAYEE
1081 M=PAYEE
1082 M=PAYEE
1083 M=PAYEE
1084 M=PAYEE
1085 M=PAYEE
1086 M=PAYEE
1087 M=PAYEE
1088 M=PAYEE
1089 M=PAYEE
1090 M=PAYEE
1091 M=PAYEE
1092 M=PAYEE
1093 M=PAYEE
1094 M=PAYEE
1095 M=PAYEE
1096 M=PAYEE
1097 M=PAYEE
1098 M=PAYEE
1099 M=PAYEE
1100 M=PAYEE
1101 M=PAYEE
1102 M=PAYEE
1103 M=PAYEE
1104 M=PAYEE
1105 M=PAYEE
1106 M=PAYEE
1107 M=PAYEE
1108 M=PAYEE
1109 M=PAYEE
1110 M=PAYEE
1111 M=PAYEE
1112 M=PAYEE
1113 M=PAYEE
1114 M=PAYEE
1115 M=PAYEE
1116 M=PAYEE
1117 M=PAYEE
1118 M=PAYEE
1119 M=PAYEE
1120 M=PAYEE
1121 M=PAYEE
1122 M=PAYEE
1123 M=PAYEE
1124 M=PAYEE
1125 M=PAYEE
1126 M=PAYEE
1127 M=PAYEE
1128 M=PAYEE
1129 M=PAYEE
1130 M=PAYEE
1131 M=PAYEE
1132 M=PAYEE
1133 M=PAYEE
1134 M=PAYEE
1135 M=PAYEE
1136 M=PAYEE
1137 M=PAYEE
1138 M=PAYEE
1139 M=PAYEE
1140 M=PAYEE
1141 M=PAYEE
1142 M=PAYEE
1143 M=PAYEE
1144 M=PAYEE
1145 M=PAYEE
1146 M=PAYEE
1147 M=PAYEE
1148 M=PAYEE
1149 M=PAYEE
1150 M=PAYEE
1151 M=PAYEE
1152 M=PAYEE
1153 M=PAYEE
1154 M=PAYEE
1155 M=PAYEE
1156 M=PAYEE
1157 M=PAYEE
1158 M=PAYEE
1159 M=PAYEE
1160 M=PAYEE
1161 M=PAYEE
1162 M=PAYEE
1163 M=PAYEE
1164 M=PAYEE
1165 M=PAYEE
1166 M=PAYEE
1167 M=PAYEE
1168 M=PAYEE
1169 M=PAYEE
1170 M=PAYEE
1171 M=PAYEE
1172 M=PAYEE
1173 M=PAYEE
1174 M=PAYEE
1175 M=PAYEE
1176 M=PAYEE
1177 M=PAYEE
1178 M=PAYEE
1179 M=PAYEE
1180 M=PAYEE
1181 M=PAYEE
1182 M=PAYEE
1183 M=PAYEE
1184 M=PAYEE
1185 M=PAYEE
1186 M=PAYEE
1187 M=PAYEE
1188 M=PAYEE
1189 M=PAYEE
1190 M=PAYEE
1191 M=PAYEE
1192 M=PAYEE
1193 M=PAYEE
1194 M=PAYEE
1195 M=PAYEE
1196 M=PAYEE
1197 M=PAYEE
1198 M=PAYEE
1199 M=PAYEE
1200 M=PAYEE
1201 M=PAYEE
1202 M=PAYEE
1203 M=PAYEE
1204 M=PAYEE
1205 M=PAYEE
1206 M=PAYEE
1207 M=PAYEE
1208 M=PAYEE
1209 M=PAYEE
1210 M=PAYEE
1211 M=PAYEE
1212 M=PAYEE
1213 M=PAYEE
1214 M=PAYEE
1215 M=PAYEE
1216 M=PAYEE
1217 M=PAYEE
1218 M=PAYEE
1219 M=PAYEE
1220 M=PAYEE
1221 M=PAYEE
1222 M=PAYEE
1223 M=PAYEE
1224 M=PAYEE
1225 M=PAYEE
1226 M=PAYEE
1227 M=PAYEE
1228 M=PAYEE
1229 M=PAYEE
1230 M=PAYEE
1231 M=PAYEE
1232 M=PAYEE
1233 M=PAYEE
1234 M=PAYEE
1235 M=PAYEE
1236 M=PAYEE
1237 M=PAYEE
1238 M=PAYEE
1239 M=PAYEE
1240 M=PAYEE
1241 M=PAYEE
1242 M=PAYEE
1243 M=PAYEE
1244 M=PAYEE
1245 M=PAYEE
1246 M=PAYEE
1247 M=PAYEE
1248 M=PAYEE
1249 M=PAYEE
1250 M=PAYEE
1251 M=PAYEE
1252 M=PAYEE
1253 M=PAYEE
1254 M=PAYEE
1255 M=PAYEE
1256 M=PAYEE
1257 M=PAYEE
1258 M=PAYEE
1259 M=PAYEE
1260 M=PAYEE
1261 M=PAYEE
1262 M=PAYEE
1263 M=PAYEE
1264 M=PAYEE
1265 M=PAYEE
1266 M=PAYEE
1267 M=PAY
```


[illegible][illegible]

Mastermind

By Tony Hinde

YOU USE the standard rules of the Mastermind board game for this program. No colours can be repeated, and the code at the end is set out at random, not in order.

Modified CP/M

By Nigel Harwood

ON BOOT-UP of my Osborne 1 computer, it automatically executes a program called AUTOST.COM, which enables any program or function to be automatically started.

This is very useful, but not always desirable. However, when I decided that it wasn't wanted for a particular disk, I ran into some problems. You see, nowhere in the manual does it mention *how* to disengage the AUTO-ST facility.

I tried simply deleting the program AUTO-ST.COM, but then CP/M booted up with the message "AUTOST?" ... which, to me, looked very poor.

The answer I found was in slightly changing the CP/M itself. Execute SYSGEN, to the source-drive prompt, enter A. To the destination-drive prompt, enter RETURN. This will then have read in the CP/M system and left it in memory. Now, enter SAVE 35 CPM.COM, which will save the first 35 pages of memory; these contain the CP/M system.

Next, execute DDT CPM.COM, then in DDT enter D2008. You should now see:

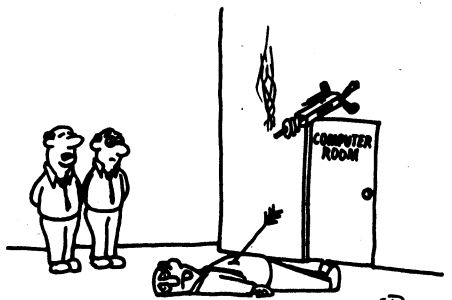
```
2008 E5 01 07 41 55 54 4F 53...AUTOS
2010 54 20 30 31 32 33 34 35 T 012345
```

Now, enter S2008 and then the following:
RETURN, RETURN, 00,20,20,20,20,20, ending by entering a fullstop and new line. Next, enter Control-C to get back to the CP/M operating system.

You have now modified the CP/M system - it was read into memory when you entered DDT, and it is still there now.

Execute SYSGEN again, answer RETL to the source-drive prompt, because you wish to use the CP/M that you have modified in memory. To the destination-drive prompt, enter A.

When the disk light has gone out, press the reset button to reboot the system. After hitting RETURN to the boot-up prompt, the improved sign-on should greet you on a cleared screen.



"LOOKS LIKE SOMEBODY FORGOT TO TELL THE ENGINEER ABOUT OUR NEW ELECTRONIC SECURITY SYSTEM."

```

90 REM ** GRAPHIC STRINGS **
100 C1:=(CHR$(150)+CHR$(184)+CHR$(191)+CHR$(180)+CHR$(144)+CHR$(
110 153)+CHR$(147)+CHR$(149)+CHR$(191)+CHR$(145)+CHR$(153)
120 C2:=(CHR$(149)+CHR$(179)+CHR$(143)+CHR$(179)+CHR$(143)+CHR$(
130 153)+CHR$(139)+CHR$(182)+CHR$(179)+CHR$(185)+CHR$(135)
140 C3:=(CHR$(154)+CHR$(198)+CHR$(178)+CHR$(188)+CHR$(118)+CHR$(
150 177)+CHR$(182)+CHR$(187)+CHR$(179)+CHR$(140)+CHR$(147)+CHR$(
160 C4:=(CHR$(156)+CHR$(140)+CHR$(182)+CHR$(151)+CHR$(179)+CHR$(
170 154)+CHR$(186)+CHR$(198)+CHR$(128)+CHR$(189)+CHR$(188)
180 C5:=(CHR$(143)+CHR$(187)+CHR$(191)+CHR$(159)+CHR$(177)+CHR$(
190 156)+CHR$(136)+CHR$(176)+CHR$(191)+CHR$(176)+CHR$(132)
200 C6:=(CHR$(128)+CHR$(120)+CHR$(160)+CHR$(182)+CHR$(187)+CHR$(
210 188)+CHR$(183)+CHR$(187)+CHR$(179)+CHR$(129)+CHR$(128)+CHR$(126)
220 C7:=(CHR$(162)+CHR$(187)+CHR$(191)+CHR$(183)+CHR$(147)+CHR$(
230 156)+CHR$(198)+CHR$(185)+CHR$(185)+CHR$(187)+CHR$(189)
240 C8:=(CHR$(140)+CHR$(179)+CHR$(140)+CHR$(179)+CHR$(140)+CHR$(
250 179)+CHR$(179)+CHR$(140)+CHR$(179)+CHR$(140)+CHR$(179)
260 C9:=(CHR$(152)+CHR$(171)+CHR$(121)+CHR$(131)+CHR$(164)+CHR$(
270 158)+CHR$(175)+CHR$(191)+CHR$(191)+CHR$(191)+CHR$(159)
280 END
500 PRINT#65,STRINGS:20,179;CHR$(187)+";PRESS 'a' TO STOP";CHR$(
170)+STRINGS(19,179);
180 GOSUB050
190 GOSUB010:PRINT#7,STRINGS(6,128);
198 Z=AND(I);X=AND(I);Y=AND(I)
200 PRINT#34,C1;Z1:=USR(1);Z10:=V1;PRINT#414,C1;X1:=USR(1);Z10:=S
210 1;PRINT#414,C2;X1:=USR(1);Z10:=V1;PRINT#414,C3;X1:=USR(1);Z10:=S
220 1;PRINT#414,C4;X1:=USR(1);Z10:=V1;PRINT#414,C5;X1:=USR(1);Z10:=S
230 1;PRINT#414,C6;X1:=USR(1);Z10:=V1;PRINT#414,C7;X1:=USR(1);Z10:=S
240 1;PRINT#414,C8;X1:=USR(1);Z10:=V1;PRINT#414,C9;X1:=USR(1);Z10:=S
250 1;PRINT#414,C10;X1:=USR(1);Z10:=V1;PRINT#414,C11;X1:=USR(1);Z10:=S
260 1;PRINT#414,C12;X1:=USR(1);Z10:=V1;PRINT#414,C13;X1:=USR(1);Z10:=S
270 1;PRINT#414,C14;X1:=USR(1);Z10:=V1;PRINT#414,C15;X1:=USR(1);Z10:=S
280 1;PRINT#414,C16;X1:=USR(1);Z10:=V1;PRINT#414,C17;X1:=USR(1);Z10:=S
290 1;PRINT#414,C18;X1:=USR(1);Z10:=V1;PRINT#414,C19;X1:=USR(1);Z10:=S
300 1;PRINT#414,C20;X1:=USR(1);Z10:=V1;PRINT#414,C21;X1:=USR(1);Z10:=S
310 1;PRINT#414,C22;X1:=USR(1);Z10:=V1;PRINT#414,C23;X1:=USR(1);Z10:=S
320 1;PRINT#414,C24;X1:=USR(1);Z10:=V1;PRINT#414,C25;X1:=USR(1);Z10:=S
330 1;PRINT#414,C26;X1:=USR(1);Z10:=V1;PRINT#414,C27;X1:=USR(1);Z10:=S
340 1;PRINT#414,C28;X1:=USR(1);Z10:=V1;PRINT#414,C29;X1:=USR(1);Z10:=S
350 1;PRINT#414,C30;X1:=USR(1);Z10:=V1;PRINT#414,C31;X1:=USR(1);Z10:=S
360 1;PRINT#414,C32;X1:=USR(1);Z10:=V1;PRINT#414,C33;X1:=USR(1);Z10:=S
370 1;PRINT#414,C34;X1:=USR(1);Z10:=V1;PRINT#414,C35;X1:=USR(1);Z10:=S
380 1;PRINT#414,C36;X1:=USR(1);Z10:=V1;PRINT#414,C37;X1:=USR(1);Z10:=S
390 1;PRINT#414,C38;X1:=USR(1);Z10:=V1;PRINT#414,C39;X1:=USR(1);Z10:=S
400 1;PRINT#414,C40;X1:=USR(1);Z10:=V1;PRINT#414,C41;X1:=USR(1);Z10:=S
410 1;PRINT#414,C42;X1:=USR(1);Z10:=V1;PRINT#414,C43;X1:=USR(1);Z10:=S
420 1;PRINT#414,C44;X1:=USR(1);Z10:=V1;PRINT#414,C45;X1:=USR(1);Z10:=S
430 1;PRINT#414,C46;X1:=USR(1);Z10:=V1;PRINT#414,C47;X1:=USR(1);Z10:=S
440 1;PRINT#414,C48;X1:=USR(1);Z10:=V1;PRINT#414,C49;X1:=USR(1);Z10:=S
450 1;PRINT#414,C50;X1:=USR(1);Z10:=V1;PRINT#414,C51;X1:=USR(1);Z10:=S
460 1;PRINT#414,C52;X1:=USR(1);Z10:=V1;PRINT#414,C53;X1:=USR(1);Z10:=S
470 1;PRINT#414,C54;X1:=USR(1);Z10:=V1;PRINT#414,C55;X1:=USR(1);Z10:=S
480 1;PRINT#414,C56;X1:=USR(1);Z10:=V1;PRINT#414,C57;X1:=USR(1);Z10:=S
490 1;PRINT#414,C58;X1:=USR(1);Z10:=V1;PRINT#414,C59;X1:=USR(1);Z10:=S
500 1;PRINT#414,C60;X1:=USR(1);Z10:=V1;PRINT#414,C61;X1:=USR(1);Z10:=S
510 1;PRINT#414,C62;X1:=USR(1);Z10:=V1;PRINT#414,C63;X1:=USR(1);Z10:=S
520 1;PRINT#414,C64;X1:=USR(1);Z10:=V1;PRINT#414,C65;X1:=USR(1);Z10:=S
530 1;PRINT#414,C66;X1:=USR(1);Z10:=V1;PRINT#414,C67;X1:=USR(1);Z10:=S
540 1;PRINT#414,C68;X1:=USR(1);Z10:=V1;PRINT#414,C69;X1:=USR(1);Z10:=S
550 1;PRINT#414,C70;X1:=USR(1);Z10:=V1;PRINT#414,C71;X1:=USR(1);Z10:=S
560 1;PRINT#414,C72;X1:=USR(1);Z10:=V1;PRINT#414,C73;X1:=USR(1);Z10:=S
570 1;PRINT#414,C74;X1:=USR(1);Z10:=V1;PRINT#414,C75;X1:=USR(1);Z10:=S
580 1;PRINT#414,C76;X1:=USR(1);Z10:=V1;PRINT#414,C77;X1:=USR(1);Z10:=S
590 1;PRINT#414,C78;X1:=USR(1);Z10:=V1;PRINT#414,C79;X1:=USR(1);Z10:=S
600 1;PRINT#414,C80;X1:=USR(1);Z10:=V1;PRINT#414,C81;X1:=USR(1);Z10:=S
610 1;PRINT#414,C82;X1:=USR(1);Z10:=V1;PRINT#414,C83;X1:=USR(1);Z10:=S
620 1;PRINT#414,C84;X1:=USR(1);Z10:=V1;PRINT#414,C85;X1:=USR(1);Z10:=S
630 1;PRINT#414,C86;X1:=USR(1);Z10:=V1;PRINT#414,C87;X1:=USR(1);Z10:=S
640 1;PRINT#414,C88;X1:=USR(1);Z10:=V1;PRINT#414,C89;X1:=USR(1);Z10:=S
650 1;PRINT#414,C90;X1:=USR(1);Z10:=V1;PRINT#414,C91;X1:=USR(1);Z10:=S
660 1;PRINT#414,C92;X1:=USR(1);Z10:=V1;PRINT#414,C93;X1:=USR(1);Z10:=S
670 1;PRINT#414,C94;X1:=USR(1);Z10:=V1;PRINT#414,C95;X1:=USR(1);Z10:=S
680 1;PRINT#414,C96;X1:=USR(1);Z10:=V1;PRINT#414,C97;X1:=USR(1);Z10:=S
690 1;PRINT#414,C98;X1:=USR(1);Z10:=V1;PRINT#414,C99;X1:=USR(1);Z10:=S
700 1;PRINT#414,C100;X1:=USR(1);Z10:=V1;PRINT#414,C101;X1:=USR(1);Z10:=S
710 1;PRINT#414,C102;X1:=USR(1);Z10:=V1;PRINT#414,C103;X1:=USR(1);Z10:=S
720 1;PRINT#414,C104;X1:=USR(1);Z10:=V1;PRINT#414,C105;X1:=USR(1);Z10:=S
730 1;PRINT#414,C106;X1:=USR(1);Z10:=V1;PRINT#414,C107;X1:=USR(1);Z10:=S
740 1;PRINT#414,C108;X1:=USR(1);Z10:=V1;PRINT#414,C109;X1:=USR(1);Z10:=S
750 1;PRINT#414,C110;X1:=USR(1);Z10:=V1;PRINT#414,C111;X1:=USR(1);Z10:=S
760 1;PRINT#414,C112;X1:=USR(1);Z10:=V1;PRINT#414,C113;X1:=USR(1);Z10:=S
770 1;PRINT#414,C114;X1:=USR(1);Z10:=V1;PRINT#414,C115;X1:=USR(1);Z10:=S
780 1;PRINT#414,C116;X1:=USR(1);Z10:=V1;PRINT#414,C117;X1:=USR(1);Z10:=S
790 1;PRINT#414,C118;X1:=USR(1);Z10:=V1;PRINT#414,C119;X1:=USR(1);Z10:=S
800 1;PRINT#414,C120;X1:=USR(1);Z10:=V1;PRINT#414,C121;X1:=USR(1);Z10:=S
810 1;PRINT#414,C122;X1:=USR(1);Z10:=V1;PRINT#414,C123;X1:=USR(1);Z10:=S
820 1;PRINT#414,C124;X1:=USR(1);Z10:=V1;PRINT#414,C125;X1:=USR(1);Z10:=S
830 1;PRINT#414,C126;X1:=USR(1);Z10:=V1;PRINT
```

APF

Disk File Fix

By J. L. Elkhorne

SOMETIMES, an erroneous disk-directory listing is the fault of one operator – but the APF microcomputer can provide a myriad of gremlins to plague a person!

If you can't remove a file to get rid of clutter on a disk, the hard way is to sequentially run-and-save the good ones on another disk and re-initialise the offender. However, as that's not very elegant, I developed the technique described here.

1. Reset system
2. Enter 10 SAVE "aaaaaaa"K
3. Enter DIR
4. CALL 28672
5. Enter D A300

A300 is the system buffer. Once DIR is done, each file is described in a 16-byte

block. Format is seven characters, followed by:

20 Delimiter

nn

nn

nn

6. Note data

c. Note data for the seven-character name pertinent to the file you wish to delete.

7. Do M A406 and replace R's (41) with data exactly.

8. Do G 8894 to take you back to BASIC.

9. Run, then DIR to ascertain that file is truly gone. ☐

MICRO DECISION



The Micro Decision is a professional, CP/M® based computer system that comes with the hardware and software needed for the large or small business. This system is ideal for the professional or small business owner who wants to enter the computer age at an amazingly low price. It offers all the word processing, financial planning and programming tools needed to increase productivity.

The big office with a backlog of work on its minis and mainframes can use the Micro Decision to generate budget projections, sales reports, resource utilization studies, off-book receivables . . . all those special projects that need to be done now, but don't justify tying up a big machine.

The Micro Decision is a single board, Z80®-based computer that comes with 64K of RAM, an on-board disk controller and one or two 5 1/4 inch floppy disk drives. An expansion port allows the Micro Decision to handle up to four 5 1/4 inch drives, while two RS-232 serial ports connect the Micro Decision to a terminal, and a printer or modem.

Standard software includes Wordstar®, the popular word processing program, and Correct-It®, a powerful spelling checker. Financial forecasting is easy with Logi-Calc®, the electronic "spreadsheet" that adds computer power to business projections. Programmers will appreciate the choice of three programming languages: Microsoft® Basic-80, BaZic®, and Morrow Designs' Pilot. The software package is completed by the CP/M 2.2 Operating System, the industry standard O/S that allows access to more than 2,000 business programs.

Morrow Designs' optional terminal offers a full 12 inch diagonal, high resolution screen with 80 characters by 24 lines. The detachable keyboard allows positioning for maximum user comfort. An extra RS-232 serial port provides opportunities for additional I/O communication. And the terminals' intelligent features significantly improve system performance.

Add a printer and you have all the tools necessary to produce and store letters, documents and financial reports.

The Micro Decision is more than just the sum total of its hardware and software specifications. Morrow Designs has added features to the Micro Decision that make it convenient for the beginner as well as the experienced CP/M user. These features include: multiple diskette formats, enhanced error handling, Virtual Drive, simple terminal configuration, extensive diagnostics, a menu-driven front end, and perhaps most important, a clearly written user manual.

The Micro Decision's straightforward component design insures continued reliability. Each system is given extensive quality control and testing before shipment.

Morrow Designs has been dedicated to high-performance, low-cost computing since 1976. First in solid state memory, then disk memory, today in both single and multi-user computer systems.

Automation Statham Pty. Ltd. has been a Morrow designs dealer since 1976.

*Plus Sales Tax if applicable

AUTOMATION STATHAM PTY. LTD.

47 Birch St, Bankstown, NSW 2200. Australia. Phone (02) 709-4144.

In Brisbane: The Softwarehouse (07) 221-9640

In Melbourne: Microtrix (03) 439-5257

Alarm Clock

By Philip Cookson

THE ALARM on this digital alarm clock can either be a "beep" alarm or a cassette on/off command, which can be programmed to record from radio or television.

```

100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****
124 *****
125 *****
126 *****
127 *****
128 *****
129 *****
130 *****
131 *****
132 *****
133 *****
134 *****
135 *****
136 *****
137 *****
138 *****
139 *****
140 *****
141 *****
142 *****
143 *****
144 *****
145 *****
146 *****
147 *****
148 *****
149 *****
150 *****
151 *****
152 *****
153 *****
154 *****
155 *****
156 *****
157 *****
158 *****
159 *****
160 *****
161 *****
162 *****
163 *****
164 *****
165 *****
166 *****
167 *****
168 *****
169 *****
170 *****
171 *****
172 *****
173 *****
174 *****
175 *****
176 *****
177 *****
178 *****
179 *****
180 *****
181 *****
182 *****
183 *****
184 *****
185 *****
186 *****
187 *****
188 *****
189 *****
190 *****
191 *****
192 *****
193 *****
194 *****
195 *****
196 *****
197 *****
198 *****
199 *****
200 *****
201 *****
202 *****
203 *****
204 *****
205 *****
206 *****
207 *****
208 *****
209 *****
210 *****
211 *****
212 *****
213 *****
214 *****
215 *****
216 *****
217 *****
218 *****
219 *****
220 *****
221 *****
222 *****
223 *****
224 *****
225 *****
226 *****
227 *****
228 *****
229 *****
230 *****
231 *****
232 *****
233 *****
234 *****
235 *****
236 *****
237 *****
238 *****
239 *****
240 *****
241 *****
242 *****
243 *****
244 *****
245 *****
246 *****
247 *****
248 *****
249 *****
250 *****
251 *****
252 *****
253 *****
254 *****
255 *****
256 *****
257 *****
258 *****
259 *****
260 *****
261 *****
262 *****
263 *****
264 *****
265 *****
266 *****
267 *****
268 *****
269 *****
270 *****
271 *****
272 *****
273 *****
274 *****
275 *****
276 *****
277 *****
278 *****
279 *****
280 *****
281 *****
282 *****
283 *****
284 *****
285 *****
286 *****
287 *****
288 *****
289 *****
290 *****
291 *****
292 *****
293 *****
294 *****
295 *****
296 *****
297 *****
298 *****
299 *****
300 *****
301 *****
302 *****
303 *****
304 *****
305 *****
306 *****
307 *****
308 *****
309 *****
310 *****
311 *****
312 *****
313 *****
314 *****
315 *****
316 *****
317 *****
318 *****
319 *****
320 *****
321 *****
322 *****
323 *****
324 *****
325 *****
326 *****
327 *****
328 *****
329 *****
330 *****
331 *****
332 *****
333 *****
334 *****
335 *****
336 *****
337 *****
338 *****
339 *****
340 *****
341 *****
342 *****
343 *****
344 *****
345 *****
346 *****
347 *****
348 *****
349 *****
350 *****
351 *****
352 *****
353 *****
354 *****
355 *****
356 *****
357 *****
358 *****
359 *****
360 *****
361 *****
362 *****
363 *****
364 *****
365 *****
366 *****
367 *****
368 *****
369 *****
370 *****
371 *****
372 *****
373 *****
374 *****
375 *****
376 *****
377 *****
378 *****
379 *****
380 *****
381 *****
382 *****
383 *****
384 *****
385 *****
386 *****
387 *****
388 *****
389 *****
390 *****
391 *****
392 *****
393 *****
394 *****
395 *****
396 *****
397 *****
398 *****
399 *****
400 *****
401 *****
402 *****
403 *****
404 *****
405 *****
406 *****
407 *****
408 *****
409 *****
410 *****
411 *****
412 *****
413 *****
414 *****
415 *****
416 *****
417 *****
418 *****
419 *****
420 *****
421 *****
422 *****
423 *****
424 *****
425 *****
426 *****
427 *****
428 *****
429 *****
430 *****
431 *****
432 *****
433 *****
434 *****
435 *****
436 *****
437 *****
438 *****
439 *****
440 *****
441 *****
442 *****
443 *****
444 *****
445 *****
446 *****
447 *****
448 *****
449 *****
450 *****
451 *****
452 *****
453 *****
454 *****
455 *****
456 *****
457 *****
458 *****
459 *****
460 *****
461 *****
462 *****
463 *****
464 *****
465 *****
466 *****
467 *****
468 *****
469 *****
470 *****
471 *****
472 *****
473 *****
474 *****
475 *****
476 *****
477 *****
478 *****
479 *****
480 *****
481 *****
482 *****
483 *****
484 *****
485 *****
486 *****
487 *****
488 *****
489 *****
490 *****
491 *****
492 *****
493 *****
494 *****
495 *****
496 *****
497 *****
498 *****
499 *****
500 *****
501 *****
502 *****
503 *****
504 *****
505 *****
506 *****
507 *****
508 *****
509 *****
510 *****
511 *****
512 *****
513 *****
514 *****
515 *****
516 *****
517 *****
518 *****
519 *****
520 *****
521 *****
522 *****
523 *****
524 *****
525 *****
526 *****
527 *****
528 *****
529 *****
530 *****
531 *****
532 *****
533 *****
534 *****
535 *****
536 *****
537 *****
538 *****
539 *****
540 *****
541 *****
542 *****
543 *****
544 *****
545 *****
546 *****
547 *****
548 *****
549 *****
550 *****
551 *****
552 *****
553 *****
554 *****
555 *****
556 *****
557 *****
558 *****
559 *****
560 *****
561 *****
562 *****
563 *****
564 *****
565 *****
566 *****
567 *****
568 *****
569 *****
570 *****
571 *****
572 *****
573 *****
574 *****
575 *****
576 *****
577 *****
578 *****
579 *****
580 *****
581 *****
582 *****
583 *****
584 *****
585 *****
586 *****
587 *****
588 *****
589 *****
590 *****
591 *****
592 *****
593 *****
594 *****
595 *****
596 *****
597 *****
598 *****
599 *****
600 *****
601 *****
602 *****
603 *****
604 *****
605 *****
606 *****
607 *****
608 *****
609 *****
610 *****
611 *****
612 *****
613 *****
614 *****
615 *****
616 *****
617 *****
618 *****
619 *****
620 *****
621 *****
622 *****
623 *****
624 *****
625 *****
626 *****
627 *****
628 *****
629 *****
630 *****
631 *****
632 *****
633 *****
634 *****
635 *****
636 *****
637 *****
638 *****
639 *****
640 *****
641 *****
642 *****
643 *****
644 *****
645 *****
646 *****
647 *****
648 *****
649 *****
650 *****
651 *****
652 *****
653 *****
654 *****
655 *****
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****
662 *****
663 *****
664 *****
665 *****
666 *****
667 *****
668 *****
669 *****
670 *****
671 *****
672 *****
673 *****
674 *****
675 *****
676 *****
677 *****
678 *****
679 *****
680 *****
681 *****
682 *****
683 *****
684 *****
685 *****
686 *****
687 *****
688 *****
689 *****
690 *****
691 *****
692 *****
693 *****
694 *****
695 *****
696 *****
697 *****
698 *****
699 *****
700 *****
701 *****
702 *****
703 *****
704 *****
705 *****
706 *****
707 *****
708 *****
709 *****
710 *****
711 *****
712 *****
713 *****
714 *****
715 *****
716 *****
717 *****
718 *****
719 *****
720 *****
721 *****
722 *****
723 *****
724 *****
725 *****
726 *****
727 *****
728 *****
729 *****
730 *****
731 *****
732 *****
733 *****
734 *****
735 *****
736 *****
737 *****
738 *****
739 *****
740 *****
741 *****
742 *****
743 *****
744 *****
745 *****
746 *****
747 *****
748 *****
749 *****
750 *****
751 *****
752 *****
753 *****
754 *****
755 *****
756 *****
757 *****
758 *****
759 *****
760 *****
761 *****
762 *****
763 *****
764 *****
765 *****
766 *****
767 *****
768 *****
769 *****
770 *****
771 *****
772 *****
773 *****
774 *****
775 *****
776 *****
777 *****
778 *****
779 *****
780 *****
781 *****
782 *****
783 *****
784 *****
785 *****
786 *****
787 *****
788 *****
789 *****
790 *****
791 *****
792 *****
793 *****
794 *****
795 *****
796 *****
797 *****
798 *****
799 *****
800 *****
801 *****
802 *****
803 *****
804 *****
805 *****
806 *****
807 *****
808 *****
809 *****
810 *****
811 *****
812 *****
813 *****
814 *****
815 *****
816 *****
817 *****
818 *****
819 *****
820 *****
821 *****
822 *****
823 *****
824 *****
825 *****
826 *****
827 *****
828 *****
829 *****
830 *****
831 *****
832 *****
833 *****
834 *****
835 *****
836 *****
837 *****
838 *****
839 *****
840 *****
841 *****
842 *****
843 *****
844 *****
845 *****
846 *****
847 *****
848 *****
849 *****
850 *****
851 *****
852 *****
853 *****
854 *****
855 *****
856 *****
857 *****
858 *****
859 *****
860 *****
861 *****
862 *****
863 *****
864 *****
865 *****
866 *****
867 *****
868 *****
869 *****
870 *****
871 *****
872 *****
873 *****
874 *****
875 *****
876 *****
877 *****
878 *****
879 *****
880 *****
881 *****
882 *****
883 *****
884 *****
885 *****
886 *****
887 *****
888 *****
889 *****
890 *****
891 *****
892 *****
893 *****
894 *****
895 *****
896 *****
897 *****
898 *****
899 *****
900 *****
901 *****
902 *****
903 *****
904 *****
905 *****
906 *****
907 *****
908 *****
909 *****
910 *****
911 *****
912 *****
913 *****
914 *****
915 *****
916 *****
917 *****
918 *****
919 *****
920 *****
921 *****
922 *****
923 *****
924 *****
925 *****
926 *****
927 *****
928 *****
929 *****
930 *****
931 *****
932 *****
933 *****
934 *****
935 *****
936 *****
937 *****
938 *****
939 *****
940 *****
941 *****
942 *****
943 *****
944 *****
945 *****
946 *****
947 *****
948 *****
949 *****
950 *****
951 *****
952 *****
953 *****
954 *****
955 *****
956 *****
957 *****
958 *****
959 *****
960 *****
961 *****
962 *****
963 *****
964 *****
965 *****
966 *****
967 *****
968 *****
969 *****
970 *****
971 *****
972 *****
973 *****
974 *****
975 *****
976 *****
977 *****
978 *****
979 *****
980 *****
981 *****
982 *****
983 *****
984 *****
985 *****
986 *****
987 *****
988 *****
989 *****
990 *****
991 *****
992 *****
993 *****
994 *****
995 *****
996 *****
997 *****
998 *****
999 *****
1000 *****

```

MBASIC Tokens

By Jeff Richards

THE TOKENS used in Microsoft BASIC have been published many times, but a program that actually finds all the BASIC reserved words, and prints them out with their tokens, may be of interest. The addresses of the routines that deal with each of the tokens are also listed.

This program was developed for Microsoft CP/M BASIC (BASIC-80) Version 5.2, but it should work for most Microsoft BASICs that use the same format.

Two variables may alter with different implementations. First, the value of BASE may change (though obviously for CP/M version it will be 100h). For ROM Basics it will probably be the base of ROM.

Secondly, the value COUNT may have to be adjusted to get the addresses correct. This value is the token of the last keyword that could validly commence a program line. In the case of BASIC 80, this is RESET

```

365 *****
366 *****
367 *****
368 *****
369 *****
370 *****
371 *****
372 *****
373 *****
374 *****
375 *****
376 *****
377 *****
378 *****
379 *****
380 *****
381 *****
382 *****
383 *****
384 *****
385 *****
386 *****
387 *****
388 *****
389 *****
390 *****
391 *****
392 *****
393 *****
394 *****
395 *****
396 *****
397 *****
398 *****
399 *****
400 *****
401 *****
402 *****
403 *****
404 *****
405 *****
406 *****
407 *****
408 *****
409 *****
410 *****
411 *****
412 *****
413 *****
414 *****
415 *****
416 *****
417 *****
418 *****
419 *****
420 *****
421 *****
422 *****
423 *****
424 *****
425 *****
426 *****
427 *****
428 *****
429 *****
430 *****
431 *****
432 *****
433 *****
434 *****
435 *****
436 *****
437 *****
438 *****
439 *****
440 *****
441 *****
442 *****
443 *****
444 *****
445 *****
446 *****
447 *****
448 *****
449 *****
450 *****
451 *****
452 *****
453 *****
454 *****
455 *****
456 *****
457 *****
458 *****
459 *****
460 *****
461 *****
462 *****
463 *****
464 *****
465 *****
466 *****
467 *****
468 *****
469 *****
470 *****
471 *****
472 *****
473 *****
474 *****
475 *****
476 *****
477 *****
478 *****
479 *****
480 *****
481 *****
482 *****
483 *****
484 *****
485 *****
486 *****
487 *****
488 *****
489 *****
490 *****
491 *****
492 *****
493 *****
494 *****
495 *****
496 *****
497 *****
498 *****
499 *****
500 *****
501 *****
502 *****
503 *****
504 *****
505 *****
506 *****
507 *****
508 *****
509 *****
510 *****
511 *****
512 *****
513 *****
514 *****
515 *****
516 *****
517 *****
518 *****
519 *****
520 *****
521 *****
522 *****
523 *****
524 *****
525 *****
526 *****
527 *****
528 *****
529 *****
530 *****
531 *****
532 *****
533 *****
534 *****
535 *****
536 *****
537 *****
538 *****
539 *****
540 *****
541 *****
542 *****
543 *****
544 *****
545 *****
546 *****
547 *****
548 *****
549 *****
550 *****
551 *****
552 *****
553 *****
554 *****
555 *****
556 *****
557 *****
558 *****
559 *****
560 *****
561 *****
562 *****
563 *****
564 *****
565 *****
566 *****
567 *****
568 *****
569 *****
570 *****
571 *****
572 *****
573 *****
574 *****
575 *****
576 *****
577 *****
578 *****
579 *****
580 *****
581 *****
582 *****
583 *****
584 *****
585 *****
586 *****
587 *****
588 *****
589 *****
590 *****
591 *****
592 *****
593 *****
594 *****
595 *****
596 *****
597 *****
598 *****
599 *****
600 *****
601 *****
602 *****
603 *****
604 *****
605 *****
606 *****
607 *****
608 *****
609 *****
610 *****
611 *****
612 *****
613 *****
614 *****
615 *****
616 *****
617 *****
618 *****
619 *****
620 *****
621 *****
622 *****
623 *****
624 *****
625 *****
626 *****
627 *****
628 *****
629 *****
630 *****
631 *****
632 *****
633 *****
634 *****
635 *****
636 *****
637 *****
638 *****
639 *****
640 *****
641 *****
642 *****
643 *****
644 *****
645 *****
646 *****
647 *****
648 *****
649 *****
650 *****
651 *****
652 *****
653 *****
654 *****
655 *****
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****
662 *****
663 *****
664 *****
665 *****
666 *****
667 *****
668 *****
669 *****
670 *****
671 *****
672 *****
673 *****
674 *****
675 *****
676 *****
677 *****
678 *****
679 *****
680 *****
681 *****
682 *****
683 *****
684 *****
685 *****
686 *****
687 *****
688 *****
689 *****
690 *****
691 *****
692 *****
693 *****
694 *****
695 *****
696 *****
697 *****
698 *****
699 *****
700 *****
701 *****
702 *****
703 *****
704 *****
705 *****
706 *****
707 *****
708 *****
709 *****
710 *****
711 *****
712 *****
713 *****
714 *****
715 *****
716 *****
717 *****
718 *****
719 *****
720 *****
721 *****
722 *****
723 *****
724 *****
725 *****
726 *****
727 *****
728 *****
729 *****
730 *****
731 *****
732 *****
733 *****
734 *****
735 *****
736 *****
737 *****
738 *****
739 *****
740 *****
741 *****
742 *****
743 *****
744 *****
745 *****
746 *****
747 *****
748 *****
749 *****
750 *****
751 *****
752 *****
753 *****
754 *****
755 *****
756 *****
757 *****
758 *****
759 *****
760 *****
761 *****
762 *****
763 *****
764 *****
765 *****
766 *****
767 *****
768 *****
769 *****
770 *****
771 *****
772 *****
773 *****
774 *****
775 *****
776 *****
777 *****
778 *****
779 *****
780 *****
781 *****
782 *****
783 *****
784 *****
785 *****
786 *****
787 *****
788 *****
789 *****
790 *****
791 *****
792 *****
793 *****
794 *****
795 *****
796 *****
797 *****
798 *****
799 *****
800 *****
801 *****
802 *****
803 *****
804 *****
805 *****
806 *****
807 *****
808 *****
809 *****
810 *****
811 *****
812 *****
813 *****
814 *****
815 *****
816 *****
817 *****
818 *****
819 *****
820 *****
821 *****
822 *****
823 *****
824 *****
825 *****
826 *****
827 *****
828 *****
829 *****
830 *****
831 *****
832 *****
833 *****
834 *****
835 *****
836 *****
837 *****
838 *****
839 *****
840 *****
841 *****
842 *****
843 *****
844 *****
845 *****
846 *****
847 *****
848 *****
849 *****
850 *****
851 *****
852 *****
853 *****
854 *****
855 *****
856 *****
857 *****
858 *****
859 *****
860 *****
861 *****
862 *****
863 *****
864 *****
865 *****
866 *****
867 *****
868 *****
869 *****
870 *****
871 *****
872 *****
873 *****
874 *****
875 *****
876 *****
877 *****
878 *****
879 *****
880 *****
881 *****
882 *****
883 *****
884 *****
885 *****
886 *****
887 *****
888 *****
889 *****
890 *****
891 *****
892 *****
893 *****
894 *****
895 *****
896 *****
897 *****
898 *****
899 *****
900 *****
901 *****
902 *****
903 *****
904 *****
905 *****
906 *****
907 *****
908 *****
909 *****
910 *****
911 *****
912 *****
913 *****
914 *****
915 *****
916 *****
917 *****
918 *****
919 *****
920 *****
921 *****
922 *****
923 *****
924 *****
925 *****
926 *****
927 *****
928 *****
929 *****
930 *****
931 *****
932 *****
933 *****
934 *****
935 *****
936 *****
937 *****
938 *****
939 *****
940 *****
941 *****
942 *****
943 *****
944 *****
945 *****
946 *****
947 *****
948 *****
949 *****
950 *****
951 *****
952 *****
953 *****
954 *****
955 *****
956 *****
957 *****
958 *****
959 *****
960 *****
961 *****
962 *****
963 *****
964 *****
965 *****
966 *****
967 *****
968 *****
969 *****
970 *****
971 *****
972 *****
973 *****
974 *****
975 *****
976 *****
977 *****
978 *****
979 *****
980 *****
981 *****
982 *****
983 *****
984 *****
985 *****
986 *****
987 *****
988 *****
989 *****
990 *****
991 *****
992 *****
993 *****
994 *****
995 *****
996 *****
997 *****
998 *****
999 *****
1000 *****

```



president

CASH & CARRY software

for IBM P.C. and Columbia M.P.C.

PERFECT WRITER:
PERFECT CALC:
PERFECT FILER:
PERFECT SPELLER:

Revolutionary Word Processor with split screen editing
Powerful Electronic Spreadsheet with 17 application programs built in
Easy to use management Data Base with mailing facility
50,000 word Spelling Dictionary, works in conjunction with Perfect Writer

FUTURE BUSINESS SYSTEMS - Powerful easy to use business package comprising of: Debtor and invoicing, Stock Control, Creditor System, General Ledger, Features Name Key Sort, General Ledger Report Formatter, Utilities, Colour, MS-DOS, Standard Stationery. Please note all features are user prompted and can be stand alone or fully integrated.

CUSTOMIZED TECHNOLOGY PRODUCTS -

PERMTER: Menu driven communications package, MS-DOS or C/PM 86
CHOMPS: Eat the monsters before they gobble you up, very popular arcade game
ISAM: Create your own data base easily with this package
GAMES PACK II: Ten great games including the old favourite SPACE TREK
TREASURE HUNT: A 3-D game where you search for the treasure in a mansion.
CATALOG: A Catalogue system that catalogues your disk and their files
CHARACTER GENERATOR: This product allows one to easily define (or redefine) graphic characters on an enlarged work tablet
CROSS REFERENCE: List variables and cross reference basic programs
ARITHMETIC TUTORIAL: This program takes one through many of the logistics of mathematics from the beginner to advanced level

QUALITY SOFTWARE - GBS DATA BASE, Menu Gen, Report Gen, Quick Gen and GBS System
Programmable relational database management system. Remarkably Powerful. CP/M 86

CARPE SOFTWARE SYSTEMS - ACCOUNTING PACKAGES includes Debtors, Stock/Invoicing, General Ledger and Creditors Systems. Also includes Couriers, Plumbers, Merchants and Clothing/Shoe trade packages, Bakers Job Costing, Manufacturing and Nursing Home Systems

SAPPHIRE MARS:

Dealers inquiries welcome



One of the most powerful menu driven super financial modelling packages CP/M 86
SYDNEY: PRESIDENT COMPUTERS 100 GEORGE ST HORNSBY •
MELBOURNE: PRESIDENT COMPUTERS 1/609 ST. KILDA RD MELBOURNE •
BRISBANE: PRESIDENT COMPUTERS 416 LOGAN RD STONES CORNER QLD •
PERTH: PRESIDENT COMPUTERS 91/1 LEURA AVE CLAREMONT W.A. •
ADELAIDE: PRESIDENT COMPUTERS 100 PIRIE STREET ADELAIDE S.A. •

PH. (02) 476 2700
(03) 529 1788
(07) 397 0888
(09) 384 5787
(08) 223 6333



president

(CCh) with only such keywords as THEN, TO and STEP having higher-valued tokens.

If you want to use this information to do some poking around in the interpreter, have a look at the TRON and TROFF routines for a little bit of very sneaky code.

```
10 DEFINT A-Z
20 BASE=&H100+5
30 COUNT=&HCC
40 C=1
50 FOR I=1 TO 27
60 IF I=27 THEN J=J+1:GOTO 90
70 I1=BASE+256+(I*2)
80 J=PEEK(I1)+PEEK(I1+1)*256
90 IS="":IF I<27 THEN IS=CHR$(I+64)
100 IF PEEK(J)=0 GOTO 240
110 IS=IS+CHR$(PEEK(J)AND &H7F)
120 IF (PEEK(J) AND &H80) GOTO 140
130 J=J+1:GOTO 110
140 K=PEEK(J+1)
150 JS="":IF K<16 THEN JS=""
160 JS=JS+HEX$(K)
170 KS="":IF K>COUNT GOTO 210
180 IF K<64 THEN K=K+COUNT
190 K=(K-128)*2+BASE
200 KS=HEX$(PEEK(K)+(PEEK(K+1)*256))
210 PRINT JS;" ";IS,KS;" ";
220 IF INT(C/3)=C/3 THEN PRINT
230 C=C+1:J=J+2:GOTO 90
240 NEXT I
```

SAMPLE RUN - MICROSOFT BASIC-80 VERSION 5.2

TK	KEYWORD	ADDR	TK	KEYWORD	ADDR	TK	KEYWORD	ADDR
F7	AND		1A	HEX\$	46B1	0E	RETURN	14FA
06	ABS	2866	85	INPUT	180D	0F	REM	1517
0E	ATN	394E	8B	IF	167A	A9	RESUME	15PC
15	ASC	4984	DA	INSTR		CA	RSET	547B
AB	AUTO	1644	05	INT	2A7F	02	RIGHT\$	499B
C3	CLOSE	53DA	10	IMP	200B	08	RND	380A
9A	COMT	4491	FB	IMP		AC	RENUM	22PC
92	CLEAR	4539	DD	INKEY\$		CC	RESET	59DA
1C	CINT	2979	C8	KILL	59D3	BB	RANDOMIZE	248A
1D	CSMG	29F3	88	LET	1541	90	STOP	4435
1E	CDBL	2A1F	B1	LINE	181F	A5	SWAP	44B4
2B	CVI	5180	C4	LOAD	52A1	CB	SAVE	539C
2C	CVS	5183	C9	LSET	547C	D4	SPEC	
2D	CVD	5186	9E	LPRINT	168D	D1	STEP	
0C	COS	38A6	9F	LLIST	289A	04	SGN	287B
16	CHR\$	4914	1B	LPOS	1D85	07	SQR	36FA
B6	CALL	4D11	93	LIST	289F	09	SIN	38AC
B8	COMMON	1515	0A	LOC	26C2	13	STR\$	46B7
B9	CHAIN	4D94	30	LOC	5644	D8	STRING\$	
84	DATA	1515	12	LEN	48F0	18	SPACES	494D
06	DIM	39A3	01	LEFT\$	496A	BD	SYSTEM	59B3
AD	DEFSTR	13FB	31	LOF	565C	A3	TROW	44AE
AE	DEFINT	13FE	C5	MERGE	5362	A4	TROFF	44AF
AF	DEFSNG	1401	FC	MOD		D0	TAB(
B0	DEFDL	1404	32	MKIS	5167	CE	TO	
98	DEF	1845	33	MKS\$	516A	CF	THEN	
AA	DELETE	2287	34	MKDS	516D	0D	TAN	3939
01	END	443A	03	MID\$	49A5	D9	USING	
A2	ELSE	1517	83	NEXT	45C8	D2	USR	
A6	ERASE	44F2	96	NULL	44A5	14	VAL	49C6
A7	EDIT	3D0E	C7	NAME	5064	DC	VARPTR	
A8	ERROR	1639	94	NEW	4354	A1	WIDTH	203C
D6	ERR		D5	NOT		97	WAIT	201C
D7	ERR		9D	OUT	2016	B4	WHILE	4C79
0B	EXP	3757	95	OW	15AB	B5	WEND	4C9C
2F	EOF	5592	BF	OPEN	50B9	B7	WRITE	5001
FA	EQV		F8	OR		F9	XOR	
82	FOR	11EE	19	OCT\$	46AB	F2	+	
C0	FIELD	5418	BA	OPTION	2440	F3	-	
C6	FILES	59FD	C2	PUT	5AF6	F4	*	
D3	FN		99	POKE	22C0	F5	/	
0F	PRE	4B18	91	PRINT	16C5	F6		
1F	FIX	2A6C	11	POS	1D89	FD	\	
89	GOTO	14C3	17	PEEK	22C1	DB	,	
89	GO TO	14C3	87	READ	1957	BF	>	
8D	GOSUB	14AB	8A	RUN	1495	F0	=	
C1	GET	5AF7	8C	RESTORE	441A	F1	<	

READ ABOUT

PRATT

DISKS FOR YOUR DGOS

APPLIED TECHNOLOGY S100 CARD OWNERS, NOW THERE IS A STANDARD IBM 8 INCH DISK SYSTEM FOR YOUR COMPUTER.
(TOTALLY ENDORSED BY APPLIED TECHNOLOGY)

SINGLE SIDED VERSION
1 x 600K DRIVE

FROM:
\$1370 + SALES TAX

SINGLE SIDED VERSION
2 x 600K DRIVE

FROM:
\$2340 + SALES TAX

DOUBLE SIDED VERSION
1 x 1.2 MEG DRIVE

FROM:
\$1804 + SALES TAX

DOUBLE SIDED VERSION
2 x 1.2 MEG DRIVE

FROM:
\$2504 + SALES TAX

(CONTROLLER ONLY IS ALSO AVAILABLE)

SYSTEMS INCLUDE CASE, SUPPLY, CABLES, MANUAL, CONTROLLER, DRIVES CP/M 2.2
MICROSOFT BASIC-80 TOTAL SOFTWARE VALUE NEAR \$600.00 FREE!

ALSO A BROAD RANGE OF OTHER S100 CARDS TO SUIT YOUR SYSTEM.



MICROCOMPUTER PRODUCTS

ACOUSTIC ELECTRONIC DEVELOPMENTS PTY., LTD. INC. IN N.S.W.

9-5
6 DAYS/WEEK
PHONE: 601-4966
TELEX: AA70664

OFFICIAL SYSTEM SUPPORT ORGANISATION FOR APPLIED TECH. S100 SYSTEMS



SHARP (AND TANDY) PCs

Spaceship Lander

By C. Colle

WITH Spaceship Lander, you can simulate the landing of a spaceship on any planet in the solar system. The table gives you the surface gravity values for the planets and some of their satellites.

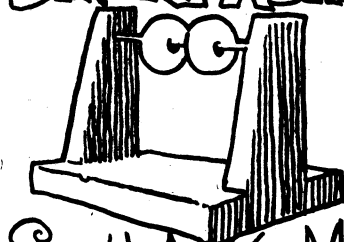
The gravity used in the sample run (1.62) is for Earth's moon, and is a good landing (soft landing and less than 200 metres from target). To have a successful landing, your speed must be less than or equal to eight metres per second when you're four metres or less above the ground.

To succeed, you have to input two accelerations, one to reduce the speed, the other to get nearer the target. If you slow down too much, the speed becomes negative, meaning that you're going up instead of down. The height, too, increases instead of decreasing. You can input an acceleration of zero, meaning free fall.

Surface Gravity Values (in m/s^2)

Mercury	3.95	Callisto	3.20
Venus	8.72	Saturn	8.77
Earth	9.81	Uranus	9.46
Moon	1.62	Neptune	13.66
Mars	3.84	Pluto	4.85
Jupiter	23.16	Ceres	0.85
Ganymede	3.43	Palas	0.54
Io	2.26	Juno	0.21
Europa	1.58	Vesta	0.43

EXCLUSIVE!
FOR VG READERS
BINARY ABACUS



SEND US ALL YOUR MONEY
& GET ONE FREE!!!

```

10:"L"PRINT "SP
    ACESHIP LAND
    ER"
15:S=1.3:D=5:M=
    2000:P=750
20:BEEP 1:INPUT
    "GRAVITY=";
    G
30:F=4*G*M:A=((
    4*F)/(3*M))-
    G:V=64*(F/M)
    :U=0:Y=(V^2)
    /(2*A):X=V
40:J=((2*Y+V^2/
    G)/(1+A/G)):
    I=(J(J/A)*P)
    /(F*S)
50:Q=M-P
55:H=INT (Y):K=
    INT (Y):L=
    INT (Q):N=
    INT (X)
60:PRINT "HEIGH
    T=";H;"SPEED
    =" ;K
70:PRINT "FUEL=
    ";L;"DISTANC
    E=";N
80:IF M=P THEN 1
    50
90:BEEP 1:INPUT
    "VERT.ACCELE
    RATION=";I;
    "HORIZ.ACCEL
    ERATION=";W
100:A=T*F/E2
110:B=W*F/E2
120:M=M-(((A+B)*
    I*D)/500)
130:IF M>P THEN 1
    50
140:A=0:B=0:M=P
150:Z=(D+(G-A/M)
    *6)
160:IF Z<0 LET Z=
    Z*-1
170:IF A=0 LET V=
    V+Z:GOTO 190
180:V=V-Z
190:U=U+D*B/M
200:Z=V*D*1.2
220:Y=Y-Z
230:Z=U*D
250:X=X-Z
260:IF Y<=4 THEN
    280
270:GOTO 50
280:IF V<=8 THEN
    310
285:GOSUB 500
290:PRINT "YOU C
    RASHED AT ";
    K;"M/S"
300:PRINT N;" M.
    FROM TARGET"
    :GOTO 380
310:IF X>200 THEN
    350
320:GOSUB 500
330:PRINT "CONGR
    ATULATIONS.
    YOU"
340:PRINT "ARE "
    ;N;" M. FROM
    TARGET":
    GOTO 380
350:GOSUB 500
360:PRINT "GOOD
    LANDING BUT"
370:PRINT N;" M.
    FROM TARGET"
380:INPUT "ANOTH
    ER GAME ? ";
    W$
390:IF W$="YES"
    THEN 15
400:END
500:N=ABS (INT (
    X))
510:RETURN
GRAVITY= 1.62
HEIGHT= 12250.
SPEED= 414.
FUEL= 1250.
DISTANCE= 414.
VERT. ACCEL.=5.
HORIZ. ACCEL.=5.
HEIGHT= 9994.
SPEED= 376.
FUEL= 1215.
DISTANCE= 406.
VERT. ACCEL.=5.
HORIZ. ACCEL.=5.

```

```

HEIGHT= 7969.
SPEED= 337.
FUEL= 1181.
DISTANCE= 389.
VERT. ACCEL.=0.
HORIZ. ACCEL.=5.
HEIGHT= 5652.
SPEED= 386.
FUEL= 1164.
DISTANCE= 364.
VERT. ACCEL.=0.
HORIZ. ACCEL.=5.
HEIGHT= 3044.
SPEED= 434.
FUEL= 1147.
DISTANCE= 331.
VERT. ACCEL.=100.
HORIZ. ACCEL.
HEIGHT= 1660.
SPEED= 230.
FUEL= 789.
DISTANCE= 287.
VERT. ACCEL.=50.
HORIZ. ACCEL.=5.
HEIGHT= 846.
SPEED= 135.
FUEL= 602.
DISTANCE= 230.
VERT. ACCEL.=50.
HORIZ. ACCEL.=5.
HEIGHT= 742.
SPEED= 17.
FUEL= 414.
DISTANCE= 160.
VERT. ACCEL.=0.
HORIZ. ACCEL.=0.
HEIGHT= 346.
SPEED= 65.
FUEL= 414.
DISTANCE= 90
VERT. ACCEL.=10.
HORIZ. ACCEL.=0.
HEIGHT= 35.
SPEED= 51.
FUEL= 380.
DISTANCE= 20.
VERT. ACCEL.=25.
HORIZ. ACCEL.=0.
CONGRATULATIONS.
YOU ARE 50. METRE
S. FROM TARGET

```

READ ABOUT

PRATTL

Mark Prediction

By C. Colle

THE MARK Prediction program will be particularly useful to students wanting to know, in advance, their average grade marks for the year.

Enter all the marks or ratings in any subject that you have received, and then those which you would like to get, and the program will tell you the percentage chance of this. However, you must assume that your work is constant. You can also find out the mean and standard deviation.

X is the mark to be entered, and Y is the frequency of this mark. When all marks have been entered, type O for the last X, then enter the mark you would like to get.

The program listing is for the printer. Press ENTER for your chances for a different mark. However, if this mark is for another subject, press SHFT M, as the memories must be cleared first.

```
10:"M"PRINT "MA
RK PREDICTIO
N":CLEAR
13:INPUT "ENTER
X ";X
15:PRINT "X = "
;X
20:IF X=0THEN 8
O
27:INPUT "ENTER
Y ";Y
30:PRINT "Y = "
;Y
40:A=YX:B=B+Y:C
=C+A:D=XX:E=
YD:F=F+E:G=C
C
70:GOTO 13
80:M=C/B
90:PRINT "MEAN
=";"USING "#
##.##";M
100:S=J((F-(G/B)
)/B)
105:Z=S/J(1-1/B)
110:PRINT "STAND
. DEVIATION=
"Z
```

```
120:V=ZZ
130:Q=0:P=0:Q=0:
R=0:T=0
137:INPUT "DESIR
ED MARK ? ";
N
140:PRINT " DESI
RED MARK ? "
;N
150:Q=N:GOSUB 20
O
160:Q=4R:Q=N-.5:
GOSUB 200
170:Q=Q+R:Q=N+.5
:GOSUB 200
180:Q=Q+R:P=6.65
Q/Z:BEEP 2
190:PRINT "YOU H
AVE ";P;" %
CHANCES":
GOTO 130
200:T=(Q-M)*(Q-M
)/V:R=J(EXP
-T):RETURN
```

```
X = 7.
Y = 1.
X = 6.
Y = 3.
X = 5.
Y = 1.
X = 0.
MEAN = 6.0
STAND. DEVIATION
= 0.7
DESIRED MARK ?
6.0
YOU HAVE 52.2 %
CHANCES
DESIRED MARK ?
7.0
YOU HAVE 22.1 %
CHANCES
DESIRED MARK ?
5.0
YOU HAVE 22.1 %
CHANCES
```

MARK PREDICTION
** SAMPLE RUN **

Alien Attack

By Scott Story

ALIEN ATTACK starts with the Alien moving along the screen to base. The attack can be checked and stopped, after five hits, by pressing the 7 button (which stops the invader when it appears on the top of the screen), the 4 button (which stops the invader in the middle of the screen) or the 1 button (which stops the invader at the bottom of the screen).

The alien then begins another attack, each time moving faster and faster as it approaches the base. You score a point each time you hit the alien.

```
1:P=1:F=0:B=40
10:"A":CLEAR:
WAIT 100:PRINT
"ALIEN ATTACK
START"
20:DIM T$(3)
30:T$(1)="0402030
204"
40:T$(2)="10080C0
810"
50:T$(3)="4020302
040"
60:GDCURSOR 110
75:CLS:P=1:F=F+1
:S=0:B=40
80:FOR I=1TO 11
```

```
90:GOSUB 5005
100:A$=INKEY$
110:IF A$="7"AND R
=1GOTO 8000
120:IF A$="4"AND R
=2GOTO 8000
130:IF A$="1"AND R
=3GOTO 8000
140:P=P+10
150:NEXT I
155:L=F*5+S-5
165:GOTO 8300
200:FOR A=0TO 20
210:BEEP 1,A
220:NEXT A
225:GDCURSOR 1
230:PAUSE "ATTACK
HALTED ":GOTO
300
300:CLS:P=1:F=F+1
:S=0:B=B-2
310:GOTO 80
5005:CLS
5007:WAIT 10:
GDCURSOR 120:
GPRINT "7F7F
7F"
5010:R=RND (3)
5020:GDCURSOR P
5030:WAIT B:
GPRINT T$(R)
:BEEP 1,60
5050:RETURN
```

```
8000:S=S+1
8005:GDCURSOR P
8007:BEEP 3,50,50
:BEEP 3,150,
100
8010:WAIT 100:
GPRINT "081C
3E1C08"
8025:IF S=5GOTO 2
00
8030:GOTO 140
8300:GDCURSOR 120
8310:GPRINT "7F7F
7F"
8400:CLS
8405:GDCURSOR 130
8420:GDCURSOR 105
8430:GPRINT T$(R)
8450:GDCURSOR 111
8455:WAIT 25:
GPRINT "0808
"
8460:GDCURSOR 111
8465:WAIT 25:
GPRINT "0808
0808"
8470:GDCURSOR 111
8475:WAIT 25:
GPRINT "0808
08080808"
8480:GDCURSOR 111
```

```
8485:WAIT 25:
GPRINT "0808
080808080808
"
8490:GDCURSOR 111
8495:WAIT 25:
GPRINT "0808
080808080808
0808"
8500:GDCURSOR 125
8510:WAIT 55:
GPRINT "2415
1E3E7F"
8515:GDCURSOR 125
8520:WAIT 60:
GPRINT "1449
2C126C7F"
8525:GDCURSOR 125
8530:WAIT 60:
GPRINT "0A20
1C282A"
8600:PAUSE "YOUR
SCORE-";L
9000:BEEP 1,90,50
:BEEP 1,70,5
0
9010:BEEP 1,150,9
0:BEEP 1,150
,100
9020:BEEP 1,50,60
:BEEP 1,250,
150
9030:END
```

COMMODORE

Ship Maths

By Ric Kube

I DEvised the Space Maths program primarily for my son to learn his addition, but it can be changed to work any of the four functions + - * and / by changing line 30.

The program gives 10 problems (minimum) and keeps score. If you get five wrong, a red alert sounds and the ship travelling across the top sinks and a new game begins. The ships and waves are made using the programmable characters. This is found in lines 30-160.

Variables:

A\$-F\$ - different parts of the ships.

G\$ - waves on sea

H\$-R\$ - erases unwanted strings to produce movement

S and CT - top left corner of screen for poking and colour

H - loop for giving problems and moving ship

SC - score

W - sums wrong

A - loop for ocean.

Some of the lines are a little long and I needed to abbreviate as much as possible.

```

5 REM**FOR VIC 20**SEE NOTES AT END
10 REM**SHIP MATHS BY RIC KUBE WAIKIEIE SA 5330 (C) 1982
20 PRINT"(clear)(black)WHAT IS YOUR NAME? INPUT N"
30 PRINT"(clear): POKE36869,255: S=7680: CT=30400
40 DATA 7168,0,0,0,0,13,13,13,13,0,0,0,128,128,128,128
50 DATA 7296,13,13,13,13,127,63,31,15,128,248,248,248,255,252,248
60 DATA 7184,129,195,103,127,127,63,63,31,128,16,56,248,224,224,248
70 DATA 7312,15,7,3,1,0,0,0,248,248,248,248,248,124,62,30
80 DATA 7280,0,0,0,0,0,1,3,0,0,0,0,64,192,192,255
90 DATA 7328,3,3,3,3,3,3,3,255,255,248,248,248,255,255
120 DATA 7216,1,3,7,15,31,63,127,255,1,3,7,15,31,63,127,255
130 DATA 7344,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
140 DATA 0
150 READ N: IF N THEN FOR N=0 TO N+5: READ A: POKE V,A: NEXT: GOTO 150
160 FOR I=7424 TO 7431: POKE I,0: NEXT
170 PRINT"(clear)"
180 A$="(down)(black)BA(down)(2 left)PO"
190 B$="(down)(black)BC(down)(2 left)RS"
200 C$="(2 down)(black)BC"
210 D$="(down)(black)DE(down)(2 left)TU"
220 E$="(2 down)(black)DE"
230 F$="(2 down)(black)DE"
240 G$="(cyan)(3 down)FD(down)(2 left)UV"
250 H$="(home)(9 down)(7 right)(9 spaces){11 left}(11 spaces)"
260 G$="(down)(2 spaces)(down)(2 left)(2 spaces)"
270 R$="(2 down)(2 spaces)"
280 PRINT"(home)": TAB(8):A$
290 FORA=0 TO 20: PRINT TAB(A):0$;NEXT
300 PRINT"(home)(9 down)(7 right)(rev)SHIP MATHS(down)(11 left)(green)
(rev)ADD THIS UP"
310 POKE36877,100: FOR L=1 TO 5: D=INT(RND(1)*5)*50+50: FOR M=370 TO 5:
POKE36878,M
320 FOR N=1700: NEXT N,M: FOR M=15 TO 3 STEP-1: POKE36878,M: FOR N=
1700: NEXT N,M,1
330 POKE36878,0: POKE36877,0
340 FOR A=0 TO 21: POKEA+3*22,230: POKECT+4*3*22,3: NEXT
350 PRINT"(home)(right)(green)(19 down)(rev)"N$
360 FOR H=0 TO 19 STEP2: PRINT"(home)":TAB(H):A$
370 PRINT H$
380 X=INT(RND(1)*12): Y=INT(RND(1)*12): Z=X*Y
390 PRINT"(home)(20 down)(rev)(blue)SCORE"SC
400 PRINT"(home)(20 down)(rev)(red)(14 right)WRONG"W
410 PRINT"(home)(11 down)(22 spaces)"
420 PRINT"(home)(11 down)(6 right)(rev)(blue)"N$+"Y"="Z": INPUTV
430 IF 0 < Z THEN 450
440 IF 0 < Z THEN 480
450 SC=SC+1: PRINT"(home)":TAB(H):0$: NEXT
460 PRINT"(clear)": POKE36878,15: FOR I=1 TO 1024: POKE36879,I:
POKE36876,I: NEXT: POKE36876,0: POKE36876,0
470 POKE36879,27: GOTO10
480 PRINT"(home)(rev)(7 down)(3 right)FOR DID IT WAS?": W$="": IF W$
THEN 560

```

```

490 PRINT"(home)(7 down)(22 spaces)"
500 GOTO 380
510 PRINT"(home)(7 down)(rev)(yellow)YOU GOT A SCORE OF: PRINT"(5 right)
(rev)(red)WITH"WRONG"
520 PRINT"(home)(11 down)(6 right)(rev)(blue)ANOTHER GO?": PRINT"(down)
(9 right)(rev)(Y/N)"
530 GET A$: IF A$="" THEN 530
540 IF A$<>"Y" THEN POKE36869,240: POKE36879,27: PRINT"(clear)(blue)":END
550 SC=0: W=0: GOTO 160
560 PRINT"(home)":FOR L=1 TO 10: FOR M=100 TO 255 STEP2: POKE36876,M:
FOR N=1700: NEXT N,M
570 POKE36876,0: FOR M=1700: NEXT M,1: POKE36876,0
580 PRINT"(home)":TAB(H):B$: FOR A=170400: NEXT: GOSUB 640
590 PRINT"(home)":TAB(H):C$: FOR A=170400: NEXT: GOSUB 640
600 PRINT"(home)":TAB(H):D$: FOR A=170500: NEXT: GOSUB 640
610 PRINT"(home)":TAB(H):E$: FOR A=170600: NEXT: GOSUB 640
620 PRINT"(home)":TAB(H):F$: FOR A=170700: NEXT: GOSUB 640
630 GOTO 510
640 PRINT"(home)":TAB(H):0$: RETURN
650 PRINT"(home)":TAB(H):R$: RETURN
700 REM ** TO FIT SOME LINES YOU NEED TO ABBREVIATE KEYWORDS **
710 REM ** SHIP MATHS ** (c) RIC KUBE WAIKIEIE SA 5330 (885) 412375
720 REM ** IF USING EXPANDED VIC CHANGE VALUE OF 'S' AND 'CT' IN 30**
730 REM ** TO 4096 AND 37888 **

```

READ ABOUT
PRATTL

G-PASCAL

The Professional Development System for *Games *Utilities *Commercial software

G-Pascal is probably the most useful program you will buy for your Apple! Its user friendliness and extremely fast compile speed (between 4,000 and 6,000 lines per minute) make it a joy to use for professionals and beginners alike. Its simplicity in fact makes it ideal for schools. The built-in text editor is very easy to use — you can actually use the Editor on its own for simple word processing applications. Operation of the compiler is very straightforward — error messages are in plain English, not numeric codes. Also available is a runtime system which will run programs independently of the compiler.

Ask your local dealer, or send cheque, money order or Bankcard number direct to us.
All orders shipped by airmail at no extra charge.

FEATURES:

- * Comprehensive Pascal subset
- * Includes compiler, editor and interpreter
- * Will run on 48K or 64K Apple with DOS 3.3
- * Support for printers and 80 column cards
- * Complete system only occupies 14K of memory
- * Includes Trace and Debug facility
- * Supports graphics, music, assembler calls
- * Powerful text editor
- * Compiles at between 4,000 and 6,000 lines per minute

G-Pascal compiler	\$85.00
Runtime system	\$35.00
Adventure game	\$35.00*
Missile Attack game (* G-Pascal source supplied)	\$35.00*
Starter Kit	
(Compiler, Runtime system plus one game)	\$110.00

Please write or phone for more information:
P.O. Box 124, Ivanhoe, Victoria, Australia 3079.
Telephone: (03) 497 1283

Gammon & Gobbett
Computer Services Pty Ltd

Symmetry

By Stewart Collins

SYMMETRY is a program that produces symmetrical patterns on a screen, and can only be stopped by pressing the space bar. It fits in one kilobyte of memory.

```

LET N=30
LET M=21
LET U=0
LET D=-1
30 LET X=RND*M-M+RND
40 LET Y=D-RND*X+RND
50 FOR R=M TO N
60 PLOT N+X,M+Y
70 LET T=X
  LET S=U
  90 LET D=D-2*(D AND S)
100 LET U=S+D
110 LET X=X*U+Y*S
120 LET Y=T*S-Y*U
130 NEXT A
140 GOTO N
  
```

Execute by typing RAND and GOTO
0. ☐

Catcher

By N. Weaver

THE IDEA of Catcher is to catch as many Os as possible with the). The program uses the RND function to set the height of the "ball" and loops to move it across the screen. The INKEY\$ function controls the height. The main part of the program is in lines 17 to 27. Line 33 decides whether you have the top score.

```

1. LET Z = 0
3. LET A$ = ""
5. PAUSE 100
7. LET A = 0
9. LET C = 14
11. CLS
13. FOR J = 1 TO 10
15. LET B = INT (RND * 19)
17. FOR I = 0 TO 15
19. PRINT AT B,I;"O";AT C,15;"")
21. IF C = B AND I = 15 THEN LET A = A+1
23. LET C = C + (INKEY $ = "6") - (INKEY $ = "7")
25. CLS
27. NEXT I
29. NEXT J
31. PRINT AT 10, 0; "SCORE =";A
33. IF A > Z THEN GOTO 39
  
```

```

35. PRINT AT 14,0;" TOP SCORE="; Z;" BY " ; A$
37. GOTO 5
39. PRINT AT 11,0; "TOP SCORE. NAME ? ";
41. INPUT A$
43. PRINT A$; " " ; A
45. LET Z = A DOWN-6
47. RAND Z UP-7
49. GOTO VAL "3"
  
```

Resistors

By Alan Hill

RESISTORS is a simple but useful program which runs on a Sinclair ZX80 with eight kilobytes of ROM and four kilobytes of RAM.

It should also run in one kilobyte, but not with the old ROM because of the arithmetic limitations.

```

10 PRINT "THIS PROGRAM CALCULATES
OVERALL RESISTANCE."
20 PRINT
30 PRINT "ARE RESISTORS IN SERIES ? (S)"
40 PRINT "OR IN PARALLEL ? (P)"
50 INPUT A$
60 IF A$="S" THEN GOTO 100
70 IF A$="P" THEN GOTO 400
80 GOTO 30
100 CLS
110 PRINT "HOW MANY RESISTORS ARE"
120 PRINT "IN SERIES TOGETHER ? ";
130 INPUT N
140 PRINT N
150 PRINT
  
```

Typing Tutor

By Peter McKay

THE AIM of Typing Tutor is to teach the user to type by being given lines five words long, to be typed as fast as possible. At the end of each line, the time is shown in seconds. Before the main program will run, program two must be run with data for the first sentences.

Sample data:

(R\$) - Jana, Rhys, Thom, Jane, Jack Scott.
(Z\$) - Threw, Found, Passed, Pushed, Helped, Sat on.
(S\$) - Fat, Big, Cold, Sick, Smug, Huge.
(O\$) - Hog, Pig, Cow, Cat, Rat, Dog.

Words of nearly the same length must be used so that spaces between the words are few. The program displays the sentence, and the last letter pushed appears in inverse. ☐

```

1000 DIM A$(6,4)
1100 DIM Z$(6,6)
1200 DIM S$(6,4)
1300 DIM O$(6,3)
1400 FOR I=1 TO 6
1500 INPUT A$(I)
1600 INPUT Z$(I)
1700 INPUT S$(I)
1800 INPUT O$(I)
1900 NEXT I
  
```

```

1 REM "TYPING TUTOR"
2 LET R=INT (RND*6)+1
3 LET Z=INT (RND*6)+1
4 LET S=INT (RND*6)+1
5 LET O=INT (RND*6)+1
6 LET M$=R$(R)+" "+Z$(Z)+" TH
E "+S$(S)+" "+O$(O)
15 PRINT AT 0,1,M$
20 FOR I=1 TO LEN M$
21 IF INKEY$="" THEN GOTO 21
22 LET A$=INKEY$
23 IF A$=CHR$ 116 THEN LET A$="
"
24 LET W=I
25 IF A$<>M$(I TO I) THEN GOTO
21
26 IF I=1 THEN GOTO 111
27 IF W=1 THEN LET W=2
28 PRINT AT 0,W-1,M$(W-1 TO W-
1)
30 PRINT AT 0,I,CHR$ (CODE M$(
I TO I)+128)
40 NEXT I
50 LET T=INT (65535-PEEK 16436
-256*PEEK 16437)/50
55 PRINT AT 1,0;"SCORE="
60 PRINT AT 1,0;"SCORE=";T
100 GOTO 1
111 POKE 16437,255
112 POKE 16436,255
113 GOTO 27
150 SAVE "TYPING TUTOR"
160 GOTO 1
  
```



```

170 PRINT "INPUT VALUES IN OHMS"
190 LET R=0
200 DIM V(N)
230 FOR J=1 TO N
240 PRINT "R";J;" ? ";
250 INPUT V(J)
260 PRINT V(J)
270 LET R=R+V(J)
280 NEXT J
290 PRINT
300 PRINT "R = ";R;" OHMS"
310 PRINT AT 20,0;"HIT NEWLINE
FOR NEXT CALCULATION"
320 INPUT Z$
330 CLS
340 GOTO 10
400 CLS
410 PRINT "HOW MANY RESISTORS
ARE"
420 PRINT "IN PARALLEL
TOGETHER ? ";
430 INPUT N
440 PRINT N
450 PRINT
460 LET RR=0
470 DIM A(N)
480 PRINT "INPUT EACH RESIST
ANCE IN OHMS"
490 FOR K= 1 TO N
510 PRINT "R";K;" ? ";
520 INPUT A(K)
530 PRINT A(K)
540 LET RR=RR+(1/A(K))
550 NEXT K
560 PRINT
570 PRINT "R = ";1/RR;" OHMS"
580 GOTO 310

```

Australia

By K.E. Johnstone

AS A TEACHER using a Sinclair ZX81 in primary grades, I've found that most programs seem to be aimed at mathematics or spelling. However, here's a truly Aussie program which has proved popular with children.

A map of Australia is printed and the user is asked to name each state. When all the states are named correctly, the name is done for the capital cities.

The program is entered in two parts – the first part sets up the map:

```

5 REM "AUSTRALIA"
10 DIM A$(132,2)
20 PRINT 1
30 FOR N=1 TO 132
40 INPUT A$(N)
50 CLS
60 PRINT N; AT 0,4;A$(N); AT 2,0;N+1
70 NEXT N
80 CLS

```

At this stage, the program should be RUN and the 132 pairs of co-ordinates entered as follows:

```

MV MU NT NS OR PR PQ PP QQ RM SL TK UJ VI VH WG XF XE XD
XC YB XA X9 W8 V7 U6 V5 U4 U3 T2 S2 R2 Q1 P2 D1 N1 M2 L2 K3
K4 J5 J6 I7 H6 H5 G5 G6 F7 E7 D7 C8 B8 A7 96 86 76 65 54 43
33 22 12 01 .1 .2 .3 .4 .5 .6 .7 .8 *9 -A *B -C -D -E -F *G
/H .I .J .K .L .M .N .O .P .Q .R .S .T .U .V .W .X .Y .Z
9T AT BU CT DT ET FU GY GT FS FR GP HQ HP IO JO KO LP LQ LR
LS LT R, R; Q/ R* P* Q/ O; O; P; Q;

```

Lines 10 to 80 may now be deleted. Remember that the map co-ordinates are now stored as A\$ so don't use RUN.

```

160 LET Z=Z+100
165 PAUSE 300
170 CLS
175 IF Z=700 THEN GOTO 90
180 PRINT TAB 3;"NAME THE CAPITAL."
190 GOTO 105
200 FOR N=1 TO 132
210 PLOT CODE A$(N,1)-19, CODE A$(N,2)-19
220 NEXT N
250 FOR N=17 TO 39 STEP 2
260 PLOT 19,N
270 NEXT N
280 FOR N=19 TO 31 STEP 2
290 PLOT N,25
300 NEXT N
310 FOR N=25 TO 13 STEP -2
320 PLOT 31,N
330 NEXT N
340 FOR N=32 TO 40 STEP 2
350 PLOT N,22
360 NEXT N
370 PLOT 41,21
371 PLOT 42,22
380 PLOT 32,16
381 PLOT 34,15
383 PLOT 36,15
384 PLOT 38,14
390 FOR N=25 TO 33 STEP 2
400 PLOT 29,N
410 NEXT N
420 PRINT AT 13,18;" "
430 RETURN
440 REM 13 ,18 = AUSTRALIAN CAPITAL TERRITORY BUT NAME TOO LONG
FOR SCREEN
510 LET B$="QUEENSLAND"
515 GOTO 1000
520 LET B$="(NEW SOUTH WALES)"
525 GOTO 1000
530 LET B$="(VICTORIA)"
535 GOTO 1000
540 LET B$="(TASMANIA)"
545 GOTO 1000
550 LET B$="(NORTHERN TERRITORY)"
555 GOTO 1000
560 LET B$="(SOUTH AUSTRALIA)"
565 GOTO 1000
570 LET B$="(WESTERN AUSTRALIA)"
575 GOTO 1000
610 LET B$="(DARWIN)"
615 GOTO 1000
620 LET B$="(BRISBANE)"
625 GOTO 1000
630 LET B$="(SYDNEY)"
635 GOTO 1000
640 LET B$="(MELBOURNE)"
645 GOTO 1000
650 LET B$="(HOBART)"
655 GOTO 1000
660 LET B$="(ADELAIDE)"
665 GOTO 1000
670 LET B$="(PERTH)"
1000 FOR M=3 TO LEN B$
1010 IF B$(M) <> " " THEN PRINT AT CODE B$(1), CODE B$(2)+M-3;"?";
1020 IF B$(M) = " " THEN PRINT " ";
1030 NEXT M
1100 INPUT C$
1110 IF C$ <> B$(3 TO) THEN GOTO 1200
1120 PRINT AT CODE B$(1), CODE B$(2)+C$;C$
1130 LET Q(R)=1
1140 GOTO 150
1200 PRINT AT 21,0;"WRONG"
1205 REM. PAUSE MAY BE USED FOR SLOWER READERS
1210 FOR N=370 TO LEN B$
1220 PRINT AT CODE B$(1), CODE B$(2)+N-3;CHR$(CODE B$(N)+128)
1230 NEXT N
1240 PRINT AT 21,0;" "

```

```

1250 LET F=F-1
2000 GOTO 150
9990 SAVE "AUSTRALIA"
9999 GOTO 90
Note line 620 - B$(1) is Graphics A.

```

Chaser

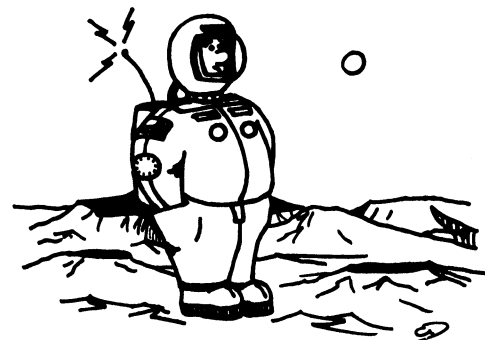
By N. Weaver

TRYING TO avoid being eaten by a monster is the aim of Chaser. When running the program, GET READY will appear, followed by a pause. Three dots will then appear on the screen, one with an H which is home, one on the left which is the monster, and the remaining dot on the right – you. The main part of the program is in lines 31 to 51.

```

1. PRINT "GET READY" (INVERSE)
3. LET Z = 1
5. LET A = INT (RND * 200)
7. LET C = INT (RND * 64)
9. PAUSE A
11. LET B = INT (RND * 44)
13. CLS
15. LET X = 63
17. LET Y = 30
19. LET L = 0
21. LET M = 30
23. PLOT X,Y
25. PLOT L,M
27. IF X = L AND M = Y THEN GOTO 57
29. PLOT C,B
30. PRINT "H"
31. IF INKEY$ = "H" THEN LET X = X + Z
33. IF INKEY$ = "C" THEN LET X = X - Z
35. IF INKEY$ = "Y" THEN LET Y = Y + Z
37. IF INKEY$ = "B" THEN LET Y = Y - Z
39. IF X = C AND Y = B THEN GOTO 65
41. IF INKEY$ = "Q" THEN LET Z = 1
43. IF INKEY$ = "A" THEN LET Z = 2
45. IF L < X THEN LET L = L + 1
47. IF L > X THEN LET L = L - 1
49. IF M < Y THEN LET M = M + 1
51. IF M > Y THEN LET M = M - 1
53. CLS
55. GOTO 15
57. CLS
59. PRINT "YOU'RE DEAD"
61. GOTO 69
63. STOP
65. CLS
67. PRINT "YOU WIN"
69. RUN

```



"BREAKER, BREAKER, THIS HERE'S THE RUBBER DUCK - YOU GOT YOUR SPAS ON, C'MON?"